DYNAMIC STABILITY CONTROL OF FRONT WHEEL DRIVE WHEELCHAIRS USING SOLID STATE ACCELEROMETERS AND GYROSCOPES

A thesis

submitted in partial fulfilment

of the requirements of the Degree

of

Master of Mechanical Engineering

in the

University of Canterbury

by

P. Wolm

University of Canterbury
2009

Dedicated to

my wife, Elizabeth

and Parents-in-Law

ABSTRACT

While the active dynamic stability of automobiles has increased over the past 17 years there have been very few similar advances made with electrically powered wheelchairs. This lack of improvement has led to a long standing acceptance of less-than-optimal stability and control of these wheelchairs. Accidents due to loss of stability are well documented. Hence, the healthcare industry has made several efforts for improved control of electric powered wheelchairs (EPWs) to provide enhanced comfort, safety and manoeuvrability at a lower cost. In response, an area of stability control was identified that could benefit from a feedback control system using solid state sensors.

To design an effective closed–loop feedback controller with optimal performance to overcome instabilities, an accurate model of wheelchair dynamics needed to be created. Such a model can be employed to test various controllers quickly and repeatedly, without the difficulties of physically setting a wheelchair up for each test. This task was one central goal of this research.

A wireless test-bed of a front wheel drive (FWD) wheelchair was also developed to validate a dynamic wheelchair model. It integrates sensors, a data control system, an embedded controller, and the motorised mechanical system. The wireless communication ensures the integrity of sensor data collected and control signals sent. The test-bed developed not only facilitates the development of feedback controllers of motorised wheelchairs, but the

collected data can also be used to confirm theories of causes of dynamic instabilities. The prototype test-bed performed the required tasks to satisfaction as defined by the sponsor.

Data collected from live tests in which the test-bed followed set patterns, was processed and analysed. The patterns were designed to induce instability. The analysis revealed that an occupied wheelchair is more stable than an unoccupied wheelchair, disproving an initial instability theory proposed in this research. However, a proximal theory explaining over-steer is confirmed.

Two models of the FWD test-bed were created. First, a dynamic model inherited from prior research, based on equations of motion was tested and enhanced based on measured data. However, even with alterations to correct parameter values and variables in the equations, a complete model validation was not possible. Second, a kinematic model was created with a factor to compensate for dynamics not normally accounted in kinematic models. The kinematic model was partially validated versus the measured data. Although, still highly accurate, there is room for improvement in this model. Both models contained a sub-system drive motor model, to account for input forces to the FWD wheelchair system model, which is fully validated.

ACKNOWLEDGEMENTS

This success of this research is attributed to a large team of people from a variety of disciplines, all helping in a vast array of areas.

My Supervisory Team: Foremost, I'd like to thank my supervisors, Dr. Geoff Chase and Dr. XiaoQi Chen from the Department of Mechanical Engineering at the University of Canterbury. Dr Chase provided the plan without which the project would never have gotten off the ground and impetus for the TIF funding that made this research possible for me. Dr Chen's enthusiasm is contagious and helped me through the more frustrating times and that Auckland conference paper was quite useful. It was important to me to know that at any stage, they were only a phone call, email or sudden site visit away.

My Company Sponsor: Many thanks to Dynamic Controls and Ian Palmer whose research initiative precipitated this project. Particular thanks to Warren Pettigrew whose enthusiasm, calm determination and invaluable knowledge moved things along. Also, Brendon Hale whose model and programming insights, and help during live tests were much appreciated.

Foundation for Research, Science & Technology: Provided the base funding for the project, but above all, has an appreciation for research.

The Technical Staff: Particularly Rodney Elliot without whom there simply would be no functioning test-bed.

Model Support: Special thanks to Dr Christopher Hann of Mechanical Engineering at the University of Canterbury for the discussions on the model and reminding me of long forgotten mathematics.

Live Testing Aide: Special thanks to also to Andi Pietzsch, the ebullient German exchange researcher who acted as my sometime camera man, more often test victim and test-bed transporter.

TABLE OF CONTENTS

1. INTRODUCTION **1**

1.1 MOTIVATION 1

1.2 PRIOR RESEARCH 3

1.2.1 The Need for Control 3

1.2.2 Modelling for Simulation and Control 4

1.2.3 Practical Implementation of Feedback Control Systems 5

1.3 SYSTEM DYNAMICS BACKGROUND 6

1.4 RESEARCH OBJECTIVES 9

1.4.1 Test-bed Development 9

1.4.2 System Modelling 11

1.4.3 Parameter Values 12

1.4.4 Control Applications 13

1.5 EQUIPMENT 13

1.5.1 Hardware 13

1.5.2 Software 14

1.6 PREFACE 16

2. WHEELCHAIR TEST-BED **19**

2.1 TEST-BED DESCRIPTION 19

2.2 WIRELESS COMMUNICATION 23

2.3 TEST-BED CONTROL DEVELOPMENT 27

2.3.1 Initial Scheme 27

2.3.2 Subsequent Changes 28

2.3.3 Current Form 31

2.4 DATA AQUISITION DEVELOPMENT 39

2.4.1 Initial Scheme 39

2.4.2 Subsequent Changes 40

2.4.3 Current Form 42

2.5 SUMMARY 46

3. DATA ANALYSIS **47**
3.1 INTRODUCTION 47
3.2 TESTING PROCESS 48
3.3 DATA PROCESSING 52
3.4 ANALYSIS 58
3.4.1 Introduction 58
3.4.2 Initial Over-Steer Theory 59
3.4.3 Disproving Initial Wheel Lift Theory 60
3.4.4 Confirmation of Initial Proximal Theory 63
3.5 SUMMARY 69

4. WHEELCHAIR SYSTEM MODELLING **71**
4.1 INTRODUCTION 71
4.2 MAIN SYSTEM MODELLING 74
4.2.1 Kinematic Equations 76
4.2.2 Dynamic Equations 77
4.3 MODEL SUB-SYSTEMS 77
4.3.1 Wheelchair Velocity 78
4.3.2 Road Load 79
4.3.3 Lateral Cornering Forces 79
4.3.4 Yaw Velocity 81
4.3.5 Position Plotting 81
4.3.6 PM DC Motor Model System 81
4.4 SUMMARY 84

5. PARAMATER VALUES **85**
5.1 INTRODUCTION 85
5.2 SYSTEM PARAMETERS 86
5.2.1 Wheelchair Mass, m 87
5.2.2 CoG Paramters, d 87
5.2.3 Mass Moment of Inertia, I 88
5.2.4 General Dimensions 90
5.3 PM DC MOTOR PARAMETERS 90
5.3.1 Torque Constant, k_t 91

5.3.2 Speed Constant, k_e 93
5.3.3 Armature Resistance, R_a 95
5.3.4 Viscous Friction Coefficient, B 95
5.3.5 Armature Inductance, L_a 95
5.3.6 Drive System Inertia, J 96
5.4 SUMMARY 97

6. MODEL ANALYSIS 99
6.1 INTRODUCTION 99
6.2 PM DC MOTOR SYSTEM MODEL ANALYSIS 99
6.2.1 Testing Process 99
6.2.2 Results 101
6.2.3 Analysis 103
6.3 FWD WHEELCHAIR MODEL SYSTEM ANALYSIS 107
6.3.1 Testing Process 107
6.3.2 Results 110
6.3.3 Analysis 117
6.4 SUMMARY 135

7. CONCLUSIONS 137
7.1 FWD WHEELCHAIR TEST-BED DEVELOPMENT 137
7.2 DATA COLLECTION 138
7.3 DATA ANALYSIS 138
7.4 MODEL VALIDATION 139
7.5 FEEDBACK CONTROL DEVELOPEMENT 140
7.6 SUMMARY 141

8. FUTURE WORK 143
8.1 FINAL MODEL DEVELOPMENT 143
8.1.1 Road Load 143
8.1.2 Mass Moment of Inertia 144
8.1.3 Over-steer Correction Factor 144

Appendix A – NEISS Sample Case Details 145

Appendix B – Control Applications 155
Appendix C – Joystick Application 157
Appendix D – PSoC Programming 159
Appendix E – CompactRIO Graphical Programming 163
Appendix F – Data Processing 167

LIST OF FIGURES

FIGURE 1.1:	INVACARE'S G-TRAC ELECTRONIC GYROSCOPE MODULE (WWW.INVACARE.COM)	6
FIGURE 1.2:	WHEELCHAIR IN OVER-STEER SITUATION (WOLM ET AL., 2006)	7
FIGURE 1.3:	TILT AND WHEEL-LIFT DUE PASSING OVER UNEVEN SURFACE AT SPEED (WOLM ET AL., 2006)	8
FIGURE 1.4:	FWD WHEELCHAIR TURNING AROUND THE LINE OF CORNING FORCE CAUSING FURTHER OVER-STEER (COLLINS AND KAUZLARICH, 1988)	8
FIGURE 1.5:	2005 PERMOBIL CHAIRMAN TEST-BED WITH CABLE TETHER (LESLIE ET AL., 2005)	10
FIGURE 1.6:	COG OF SEATED HUMAN BODY	12
FIGURE 1.7:	DSPACE REAL-TIME HARDWARE PLATFORM (WOLM ET AL., 2006)	14
FIGURE 1.8:	SIMULINK MODEL TO SEND CONTROL SIGNALS AND RECEIVE DATA TO BE LOADED ONTO THE DSPACE PLATFORM	15
FIGURE 1.9:	CONTROLDESK GUI PAIRED WITH THE LOADED SIMULINK MODEL FOUND IN FIGURE 1.8	16
FIGURE 2.1:	FRONT VIEW OF PERMOBIL CHAIRMAN TEST-BED	20
FIGURE 2.2:	SIDE VIEW OF PERMOBIL CHAIRMAN TEST-BED	20
FIGURE 2.3:	TEST-BED CHASSIS CONTAINING BATTERIES AND PM DC DRIVE MOTORS	21
FIGURE 2.4:	GEAR BOX AND PM DC DRIVE MOTOR SHOWING A SINGLE ATTACHMENT POINT TO THE CHASSIS (LESLIE ET AL., 2005)	22
FIGURE 2.5:	TYPICAL EP WHEELCHAIR GEAR BOX WITH M INDICATING WHERE A DRIVE MOTOR IS ATTACHED	22
FIGURE 2.6:	BRAINBOX BL-819 BLUETOOTH WIRELESS RS232 DONGLE (WWW.BRAINBOXES.COM)	24
FIGURE 2.7:	SPARKFUN BLUESMIRF RP-SMA BLUETOOTH WIRELESS DONGLE (WWW.SPARKFUN.COM)	25
FIGURE 2.8:	SPARKFUN FIREFLY BLUETOOTH WIRELESS RS232 DONGLE (WWW.SPARKFUN.COM)	25
FIGURE 2.9:	BASE COMMUNICATION HUB WITH 2 CHANNELS	26
FIGURE 2.10:	SEPARATE TEST-BED BLUETOOTH COMMUNICATION DONGLES	27
FIGURE 2.11:	ANALOGUE JOYSTICK CONTROLLER FROM DYNAMIC CONTROLS (LESLIE ET AL., 2005)	28

FIGURE 2.12: SHARK POWER MODULE WITH ATTACHED JOYSTICK AND CABLES FOR BATTERY ATTACHMENT 29

FIGURE 2.13: WHEELCHAIR TEST-BED EQUIPMENT OVERLOAD WITH PC 'BUFFER' (WOLM ET AL., 2006) 30

FIGURE 2.14: REQUIRED DATA PATH FOR CONTROL INSTRUCTIONS (WOLM ET AL., 2006) 30

FIGURE 2.15: ROBOTEQ AX2550 DIGITAL DC MOTOR CONTROLLER (WWW.ROBOTEQ.COM) 31

FIGURE 2.16: R/C PWM INPUT INTERPRETER CIRCUIT PCB, 50.6 MM W X 51.6 MM L X 15 MM MAX.H 32

FIGURE 2.17: GENERAL SQUARE WAVE 33

FIGURE 2.18: LOGITECH WINGMAN EXTREME DIGITAL JOYSTICK CONTROL (WWW.LOGITECH.COM) 35

FIGURE 2.19: PSOC MICROCONTROLLER 24 PIN CHIP AS FOUND ON THE INTERPRETER CIRCUIT PCB 36

FIGURE 2.20: CYPRESS SEMICONDUCTOR CORPORATION'S PSOC DESIGNER GUI 36

FIGURE 2.21: CONTORL COMMUNICATION CHAIN 38

FIGURE 2.22: DATA ACQUISITION WITH TOO MANY CABLES AND EQUIPMENT (LESLIE ET AL., 2005) 39

FIGURE 2.23: TEST-BED SENSORS AND THEIR LOCATIONS (WOLM ET AL., 2006) 40

FIGURE 2.24: TEST-BED SENSOR DATA COLLATION AND POWER SUPPLY DEVICE (WOLM ET AL., 2006) 42

FIGURE 2.25: COMPACTRIO EMBEDDED CONTROL AND DATA ACQUISITION SYSTEM (WWW.NI.COM) 44

FIGURE 2.26: LABVIEW GUI SHOWING DATA ACQUISITION CONTROL PROJECT THAT WAS LOADED ONTO COMPACTRIO 45

FIGURE 3.1: FLAT-S PROFILE KNOWN AS A DOG LEG PATTERN 48

FIGURE 3.2: PLAN VIEW OF OVAL PATTERN WHEELCHAIR TEST DRIVE TRACE 49

FIGURE 3.3: PROCESSED DATA THAT HAS BEEN CONVERTED TO WHEEL VELOCITIES AND OVERALL WHEELCHAIR VELOCITY WITH AN AREA OF NO DATA, FOLLOWED BY A DELAY IN DATA RESPONSE EVEN WITH CONTROL INPUTS OVER 60% 50

FIGURE 3.4: JOYSTICK INTERPRETER SUB-SYSTEM FOUND IN SIMULINK CONTROL MODEL THAT READS INPUTS FROM THE LOGITECH JOYSTICK VIA CONTROLDESK/DSPACE 50

FIGURE 3.5: ISOLATED VIEW OF THE STEERINGCONTROLLER VIRTUAL INSTRUMENT IN CONTROLDESK/DSPACE 51

FIGURE 3.6: CONTROL MODEL WITH INPUTS PROVIDED FROM A SAVED .MAT FILE REPLACING THE JOYSTICK INTERPRETER 51

FIGURE 3.7: WHEEL LIFT DUE TO 'MOMENTUM' FORCE CAUSING OVER-STEER (WOLM ET AL., 2006) 60

FIGURE 3.8: CALCULATED WHEEL LIFT HEIGHTS FROM INTEGRATED ROLL RATE DATA 61

FIGURE 3.9: PLAN VIEW OF ACTUAL DRIVEN GROUND TRACK OF DOG LEG RIGHT TEST PROFILE 62

FIGURE 3.10: COMPARISON OF YAW RATE WITH AND WITHOUT OCCUPIER FOR SAME CONTROL INPUTS 63

FIGURE 3.11: PLAN VIEW OF ACTUAL DRIVEN GROUND TRACK OF DOG LEG RIGHT PROFILE DRIVEN IN REVERSE 64

FIGURE 3.12: PLAN VIEW OF ACTUAL DRIVEN GROUND TRACK OF DOG LEG RIGHT PROFILE DRIVEN FORWARD WITH LOSS OF CONTROL 65

FIGURE 3.13: PATTERN 1 - YAW WITH VELOCITY FOR DOG LEG RIGHT PROFILE DRIVEN FORWARD WITHOUT LOSS OF CONTROL 66

FIGURE 3.14: PATTERN 2 - YAW WITH VELOCITY FOR DOG LEG RIGHT PROFILE DRIVEN IN REVERSE 67

FIGURE 3.15: PATTERN 3 - YAW WITH VELOCITY FOR DOG LEG RIGHT PROFILE DRIVEN FORWARD WITH LOSS OF CONTROL 67

FIGURE 4.1: FWD SIMULINK MODEL INHERITED FROM 2006 RESEARCH GROUP 73

FIGURE 4.2: FREE BODY DIAGRAM OF RWD WHEELCHAIR (DING ET AL., 2004) 74

FIGURE 4.3: BLOCK IN 2006 SIMULINK MODEL WHERE FINAL DERIVATION OF LATERAL FORCES ON ALL WHEELS CALCULATED 80

FIGURE 4.4: SIMPLE DC DRIVE MOTOR CIRCUIT 82

FIGURE 5.1: LOAD CELL MASS MEASURING SYSTEM (WOLM ET AL., 2006) 87

FIGURE 5.2: FINDING COG THROUGH LEVERAGE (WOLM ET AL., 2006) 88

FIGURE 5.3: USING QUADRIFILAR PENDULUM TO FIND WHEELCHAIR TEST-BED MOMENT OF INERTIA (WOLM ET AL., 2006) 89

FIGURE 5.4: TEST CONFIGURATION TO MEASURE MOTOR TORQUE CONSTANT (WOLM ET AL., 2006) 92

FIGURE 5.5: RESULTS OF TORQUE VERSUS CURRENT TEST, WHERE K_T = SLOPE 93

FIGURE 5.6: STEADY STATE RESULTS OF EQUATION (5.5) WHERE K_E = SLOPE 94

FIGURE 6.1: MEASURED STEADY STATE VALAUES OF PM DC DRIVE MOTOR SYSTEM FOR GIVEN VOLTAGE INPUTS (WOLM ET AL., 2008) 100

FIGURE 6.2: ACTUAL CURRENT MEASURED COMPARED WITH EQUIVALENT MODEL OUTPUT (WOLM ET AL., 2008) 101

FIGURE 6.3: ACTUAL ROTATIONAL VELOCITY MEASURED COMPARED WITH EQUIVALENT MODEL OUTPUT (WOLM ET AL., 2008) 103

FIGURE 6.4: ENHANCED DC DRIVE MOTOR CIRCUIT (WOLM ET AL., 2008) 104

FIGURE 6.5: NEW PM DC MOTOR SYSTEM SIMULINK MODEL INCORPORATING T_C 105

FIGURE 6.6: ACTUAL CURRENT MEASURED COMPARED WITH EQUIVALENT ENHANCED MODEL OUTPUT (WOLM ET AL., 2008) 106

FIGURE 6.7: ACTUAL ROTATIONAL VELOCITY MEASURED COMPARED WITH EQUIVALENT ENHANCED MODEL OUTPUT (WOLM ET AL., 2008) 107

FIGURE 6.8: LATERAL ACCELERATIONS AT RIGHT AND LEFT DRIVE WHEELS VERSUS YAW AROUND VERTICAL AXIS 109

FIGURE 6.9: LONGITUDINAL ACCELERATIONS AT RIGHT AND LEFT DRIVE WHEELS VERSUS YAW AROUND VERTICAL AXIS 109

FIGURE 6.10: MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 0.871 KGM^2 110

FIGURE 6.11: MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 5.5 KGM^2 111

FIGURE 6.12: MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL WITH I = 0.871 KGM^2 111

FIGURE 6.13: MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL WITH I = 5.5 KGM^2 112

FIGURE 6.14: MODEL VS ACTUAL FORWARD VELOCITY FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 0.871 KGM^2 112

FIGURE 6.15: MODEL VS ACTUAL FORWARD VELOCITY FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 5.5 KGM^2 113

FIGURE 6.16: MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 0.871 KGM^2 114

FIGURE 6.17: MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 5.5 KGM^2 114

FIGURE 6.18: MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL WITH I = 0.871 KGM^2 115

FIGURE 6.19: MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL WITH I = 5.5 KGM^2 115

FIGURE 6.20: MODEL VS ACTUAL YAW FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 0.871 KGM^2 116

FIGURE 6.21: MODEL VS ACTUAL YAW FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 5.5 KGM^2 116

FIGURE 6.22: ROBOTEQ MOTOR CONTROLLER OUTPUT VOLTAGE IN RESPONSE TO GIVEN DUTY CYCLE DEMANDS 118

FIGURE 6.23: NEW MAIN PLANT DYNAMICS BLOCK IN DYMANIC FWD SYSTEM MODEL 120

FIGURE 6.24: CHANGED MODEL VS ACTUAL FORWARD VELOCITY FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL 121

FIGURE 6.25: CHANGED MODEL VS ACTUAL YAW FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL 121

FIGURE 6.26: ICC FOR DIFFERENTIAL DRIVE MOBILE ROBOT AND A FWD WHEELCHAIR (SIEGWART AND NOURBAKHSH, 2004) 122

FIGURE 6.27: FINDING R FROM THE ICC FOR A DIFFERENTIAL DRIVE ROBOT OR FWD WHEELCHAIR (HAN ET AL., 2008) 123

FIGURE 6.28: ALTERED MODEL VS ACTUAL FORWARD VELOCITY FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL 124

FIGURE 6.29: ALTERED MODEL VS ACTUAL YAW FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL 124

FIGURE 6.30: KINEMATIC FWD WHEELCHAIR SYSTEM MODEL 126

FIGURE 6.31: KINEMATIC MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL 127

FIGURE 6.32: KINEMATIC MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL 127

FIGURE 6.33: KINEMATIC MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL 128

FIGURE 6.34: KINEMATIC MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL 128

FIGURE 6.35: KINEMATIC MODEL VS ACTUAL FORWARD VELOCITY FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL 129

FIGURE 6.36: KINEMATIC MODEL VS ACTUAL YAW FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL 129

FIGURE 6.37: KINEMATIC MODEL DRIVE WHEEL ROTATIONAL VELOCITIES FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL 131

FIGURE 6.38: KINEMATIC MODEL DRIVE WHEEL ROTATIONAL VELOCITIES FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL 132

FIGURE 6.39: KINEMATIC MODEL DRIVE WHEEL ROTATIONAL VELOCITIES FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL 132

FIGURE B.1: INCORPORATION OF FWD WHEELCHAIR SYSTEM MODEL INTO EMULATOR FOR FEEDBACK CONTROL DESIGN 155

FIGURE C.1: RANGE OF MOTION OF JOYSTICK IN X AND Y AXIS 157

FIGURE E.1: REAL TIME VI THAT COLLATES SENSOR DATA FOR SERIAL TRANSMISSION FROM COMPACTRIO 164

FIGURE E.2: FPGA VI SHOWING THE WHILE LOOPS THAT PROCESS THE ENCODER AND GYRO DATA BEFORE BEING PASSED TO THE REAL TIME VI 165

FIGURE E.3: FPGA VI SHOWING THE WHILE LOOPS THAT PROCESS THE ACCELEROMETER DATA BEFORE BEING PASSED TO THE REAL TIME VI 166

LIST OF TABLES

TABLE 1.1: DSPACE MODULAR SET-UP FOR PROJECT 14

TABLE 2.1: BASIC PERMOBIL CHAIRMAN WHEELCHAIR DIMENSIONS 19

TABLE 2.2: MAKEUP OF TRANSMITTED 122 BYTE DATA STREAM 41

TABLE 2.3: DATA SENSORS USED IN THE RESEARCH 43

TABLE 2.4: RESEARCH COMPACTRIO MODEL AND CONFIGURATION 44

TABLE 3.1: NUMBER OF DYNAMIC TESTS AND CONDITIONS 49

TABLE 3.2: CONVERTING A BINARY VALUE TO FLOATING POINT 56

TABLE 3.3: DOG LEG RIGHT TEST PROFILES' CONTROL INPUTS FOR THE INITIAL TURN 68

TABLE 5.1: SYSTEM PARAMETER VALUES 86

TABLE 5.2: PM DC MOTOR SYSTEM PARAMETER VALUES 91

TABLE 6.1: MSE BETWEEN MOTOR MODEL AND ACTUAL MEASURED OUTPUTS 107

TABLE 6.2: KINEMATIC MODEL STEP INPUTS FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL 133

TABLE 6.3: KINEMATIC MODEL STEP INPUTS FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL 134

TABLE 6.4: KINEMATIC MODEL STEP INPUTS FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL 134

1

INTRODUCTION

1.1 MOTIVATION

There have been numerous studies and attempts to improve the stability of manual and electric powered (EP) wheelchairs. Kirby et al. [13-15] approached the issue of stability, and its influence on performance and handling through mechanical means. Models regarding analysis and control of both manual and EP wheelchairs were also starting to be developed in parallel. For instance, Cooper [4] created a model to analyse and control the dynamic roll stability of manual, 3 or 4 wheeled wheelchairs. Brown et al. [2] developed a model for a proportional, integral, derivative (PID) controller implemented on a microprocessor, which is now an industry standard for joy-stick controllers of EP wheelchairs. Brown et al.'s controller did not overcome instabilities per se, but provided better management, handling and control overall. In addition, its PID controllers' parameters were pre-set and could not be adjusted to changes in motion of wheelchair dynamics.

However, the majority of studies focused on rear wheel drive (RWD) wheelchairs. In a study by Gaal et al. [9], 109 wheelchair riders reported 253 adverse incidents with 106

'Tips and Falls' while travelling primarily over paved streets and sidewalks in the San Francisco area. A *Tip* was defined where one or more wheels leave the ground, but the wheelchair may or may not tip over. A *Fall* is where the rider falls out of the wheelchair with no reported tipping. In each such incident where an EP wheelchair was in use, the same RWD type was involved. RWD wheelchairs accounted for 25 tips/falls that occurred driving up, down or sideways on slopes, over rough terrain, or over relatively small step changes in height. Two further incidents occurred while travelling straight on level even surfaces.

In a further study, Rentschler et al. [21] looked at 5 different EP wheelchair types; 3 RWD, 1 mid-wheel drive (MWD) and 1 front wheel drive (FWD). These were then subjected to a series of tests. Two tests in particular which would have a significant bearing on driving stability and safety indicated that FWD wheelchairs had an advantage in most aspects. The first, the angle at which a wheelchair would tilt in static tests, showed that the FWD wheelchair was superior. Second, in dynamic tests driving up and down slopes in both forward and reverse, the FWD wheelchair had some of the best stability characteristics. The exception to this latter outcome occurred when suddenly braking while driving forward on a down slope.

Certain studies by Collins and Kauzlarich [3] and other observations [7, 11] showed that of the three main drive types; front, rear and mid-wheel drive, FWD wheelchairs provide high manoeuvrability in confined spaces comparable to MWD ones, are considered to have a better fit for the users under tables or desks, have better curb climbing ability, and are the

best over uneven or sloped terrain than the other types. However, FWD wheelchairs suffer from an inherent dynamic instability due to 'over-steer' or 'veering' [11]. Hence, this single issue makes them the least comfortable type for the typical powered wheelchair user.

Thus, this specific stability and handling issue may be the one significant reason FWD wheelchairs have less of a market share than RWD and MWD ones, aside of reasons of tradition. Given the numerous advantages of FWD wheelchairs, it is clear an advantage would be gained for both wheelchair users and manufacturers by providing advanced control to overcome their main drawback. Secondly, automated control would potentially minimize costs by avoiding more expensive mechanical design solutions.

1.2 PRIOR RESEARCH

1.2.1 The Need for Control

In the United Sates alone, based on 1994-1995 data, there where an estimated 1.7 million users of wheelchairs or scooters [19]. By 2005 there were approximately 3.3 million [26]. It is clear that the number of EP wheelchair users has increased and with it the number of accidents involving wheelchair users. According to the US Consumer Product Safety Commission's National Electronic Injury Surveillance System (NEISS) 1,511 cases of wheelchair related injuries were recorded in 1997 and 2,943 in 2007, almost a two-fold increase in one decade. NEISS takes a set sample of a number of hospital's reports in the US and its territories and creates a national estimate from the numbers, hence nationally it was estimated there were 70,112 wheelchair related accidents in 1997 and 118,901 in 2007.

Corfman et al. [5] looked at 4 different EP wheelchairs; 3 RWD and 1 MWD, stating these "were selected on the basis of common use…", and, "The EPWs were driven over 4 commonly encountered obstacles". Out of 432 trials for each EPW, each manoeuvring over or on the obstacles, there were 85 losses of control (instability) and 9 falls.

Looking at a sample of NEISS case reports for wheelchair related accidents in 2007 (Appendix A) it appears that the majority of reports indicate a loss control although not explicitly stated otherwise. In light of Corfman et al.'s study and the NEISS records, a means of stability control would reduce the discomfort and associated medical costs by possible up to 25% worldwide.

1.2.2 Modelling for Simulation and Control

Before any control design of a system can be attempted, a model of the system's dynamics is required. To that end, there have been diverse system models of EP wheelchairs created and then simulated for validation experiments to understand their dynamics [1,3,16]. However, these models are based on RWD designs and are restricted to analysing the specific situations for which the mathematical equations were derived.

Ding et al. [6] have proven to be an exception noting that "…the study of reverse driving of wheelchairs with front castors is also useful for studying the forward driving of wheelchairs with rear castors…". Nevertheless, Ding et al. were concerned mainly with the effects of castor angle at the onset of backward, or forward in the case of FWD, driving.

In any case, the trend and type of wheelchair model dynamic simulations appears to be growing. Pinkney [20] has created a simulation with visual animation for a MWD wheelchair. Fattouh et al. [8] devised an animated 3D simulation of a FWD wheelchair but have not verified the model.

1.2.3 Practical Implementation of Feedback Control Systems

The industry standard for EP wheelchair control has been outlined by Brown et al. [2]. These controllers are primarily concerned with taking an input from a user via a joystick and then interpreting those inputs to send the correct voltages to the driving motors. The controllers also comprise other types of control, such as 'load compensation', so a wheelchair can maintain driving force and speed up a reasonable incline. However, all the diverse types of control within a single joystick controller can be considered internal, concerned only with the performance of the driving motors in given situations. They do not cover instabilities due to wheelchair dynamics but any further control will need to be incorporated within existing joystick controllers.

As a result of studies of wheelchair instabilities and of modelling for simulation and control, there has been a response by wheelchair manufacturers to implement stability control. A patent by Sunrise Medical [25] undertook to create an active stability control of wheelchairs based on positive angle sensors. However, at this time it appears that it has not been included in their wheelchair products.

In contrast, a patent lodged by Invacare [12] has resulted in the "G-Trac" electronic gyroscope module (Figure 1.1). It appears to be applicable only to RWD wheelchairs, designed particularly for a known instability while driving on side sloped or changing terrain. According to Invacare, G-Trac "…improves both tracking performance and traction – without adding expensive Gearless Brushless GB^{TM} motors" [12]. G-Trac is available as an option that can be added to Invacare's MK6i controller.

FIGURE 1.1: INVACARE'S G-TRAC ELECTRONIC GYROSCOPE MODULE (WWW.INVACARE.COM)

Both the Sunrise Medical and Invacare patents show that there is an interest in bringing stability control via solid state sensors to wheelchairs. In particular, Invacare has been able to implement a working system. However there, is still a need for further forms of stability control for other wheelchair drive types.

1.3 SYSTEM DYNAMICS BACKGROUND

Over-steer occurs when a vehicle turns tighter than the intended path (Figure 1.2). It is in effect a deviation from the desired path or direction as input by the wheelchair user. Once a

wheelchair is in an over-steer situation, it can lead to greater instability as an (often uncoordinated) user tries to compensate. The result of this dynamic behaviour can be either that the wheelchair spins or oscillates violently. In severe cases, it can result in throwing the typical user out of the chair, or, if in a confined situation such as in a corridor, possibly impacting a wall or other object.

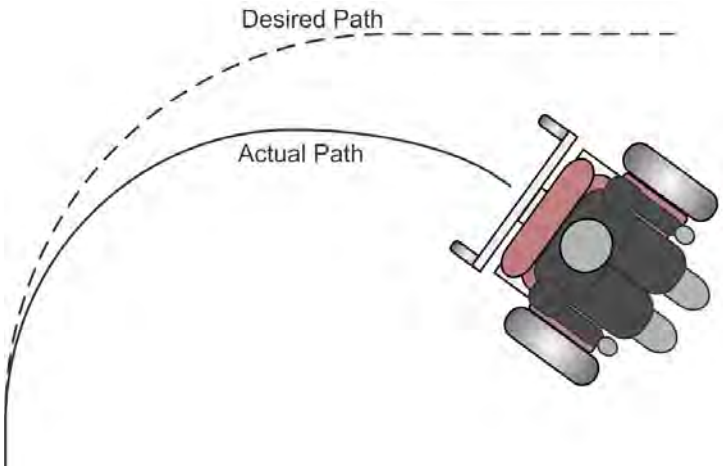

FIGURE 1.2: WHEELCHAIR IN OVER-STEER SITUATION (WOLM ET AL., 2006)

The main cause of over-steer in FWD wheelchairs can be put down to the directional instability of the rear 'free floating' caster wheels employed to aid manoeuvrability [3]. This design is a recurrent problem in this specific type of incident and in general. Incident-wise, while travelling at forward speeds and suddenly passing over an uneven surface, wheel-lift can occur, as shown in Figure 1.3. The wheel still in contact with the ground provides a relatively high differential driving force. In combination with the lack of a stabilising lateral force from the 'free floating' casters, this force imbalance results in an over-steer condition. The same result will occur if only one drive wheel loses traction such as may occur on a wet surface. The over-steer effect is much greater in either of these scenarios with higher velocities.

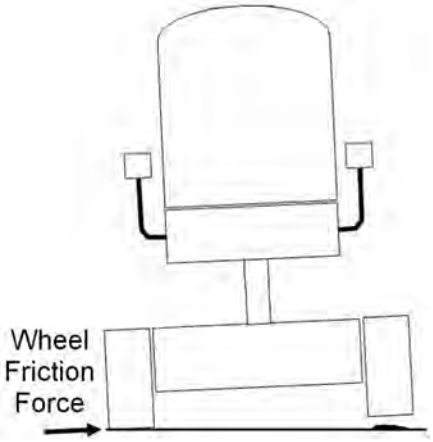

FIGURE 1.3: TILT AND WHEEL-LIFT DUE PASSING OVER UNEVEN SURFACE AT SPEED (WOLM ET AL., 2006)

Generally, even with all wheels firmly on the driving surface and with no loss of traction, the sharpness of turns are exacerbated by the fact that the lateral cornering forces act ahead of the centre of gravity (CoG) when viewed from above, in Figure 1.4.

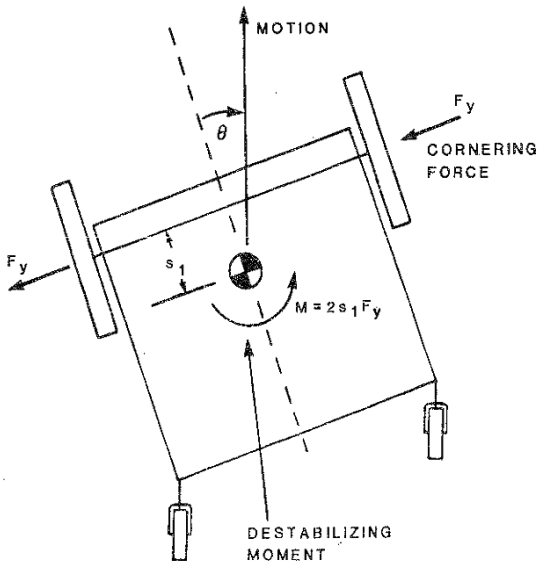

FIGURE 1.4: FWD WHEELCHAIR TURNING AROUND THE LINE OF CORNING FORCE CAUSING FURTHER OVER-STEER (COLLINS AND KAUZLARICH, 1988)

The resulting moment, M, causes the wheelchair to rotate further than desired from the commanded direction. With a lateral road cornering force magnitude of $2F_y$, it can be observed that the farther the CoG is located behind the pivot line of lateral road force (distance s_1), the more noticeable the over-steer. Again, it should be noted that, the rear casters provide no counter to this condition.

1.4 RESEARCH OBJECTIVES

The overall objective of this research is to provide a valid model of FWD wheelchair dynamics and use this model to design a closed loop feedback control with the aid of solid sate sensors to manage over-steer instability. The research presented focuses on:

1. Fully developing a FWD wheelchair test-bed
2. Obtaining useful data from the test-bed for model verification
3. Analysing the data to confirm or clarify instability theories
4. Verifying the system model
5. Applying the system model to controller development

1.4.1 Test-bed Development

In 2005, Dynamic Controls, a Christchurch, NZ based manufacturer of EP wheelchair electronic joystick controllers, provided a University of Canterbury project team with a Permobil Chairman EP FWD wheelchair, circa 1995 model (Figure 1.5). The project team attempted to obtain useful dynamic data from sensors and the analogue joy-stick controller

on the wheelchair with live tests [17]. It was found that there were large amounts of signal interference and noise through a data cable 'tether' that required extensive filtering, which was only partially successful.

FIGURE 1.5: 2005 PERMOBIL CHAIRMAN TEST-BED WITH CABLE TETHER (LESLIE ET AL., 2005)

The following year, a new University team continued the project and a decision was made to obtain all data and send control inputs through a wireless connection [28]. This approach had the advantage of eliminating signal interference and noise, and also enabled a range of travel and motion that was not permissible with the cable 'tether'. More sensors were added and the analogue joy-stick controller was replaced with a digital controller, also sourced from Dynamic Controls. Nonetheless, due to timing issues between the wireless

devices and the joy-stick controller necessitating further equipment complexity, reliable control was not achieved.

1.4.2 System Modelling

A first model was developed (in 2006), derived from Newtonian principles and trigonometry, and based on equations presented by Ding et al. [6]. The coupled equations for the electric motors, the prime sources of system driving force, were initially obtained from Rizzoni [22]. Hence, the model comprises both electrical drive motor and overall wheelchair dynamics, but does not include the effects of rear caster motion or shifting of the CoG caused by the movement of a typical user in a wheelchair, which is often due to these users' lack of (complete) muscle control. Essentially, this first model is a 2-D representation. However, it is able to obtain output in a third plane, such as roll about the longitudinal axis, or the effects of driving in a third plane such as on an incline.

A human body model was also required to understand how much it would change the CoG location of an occupied wheelchair. The 2006 project team had developed a model but the model could not be verified during the current research as the data was unavailable. Hence, further Anthropometric or human body measurement data was required [18] and another model was developed as seen in Figure 1.6 to find a seated body's CoG location. Thus, the change in the location of the occupied wheelchair test-bed's CoG could be calculated.

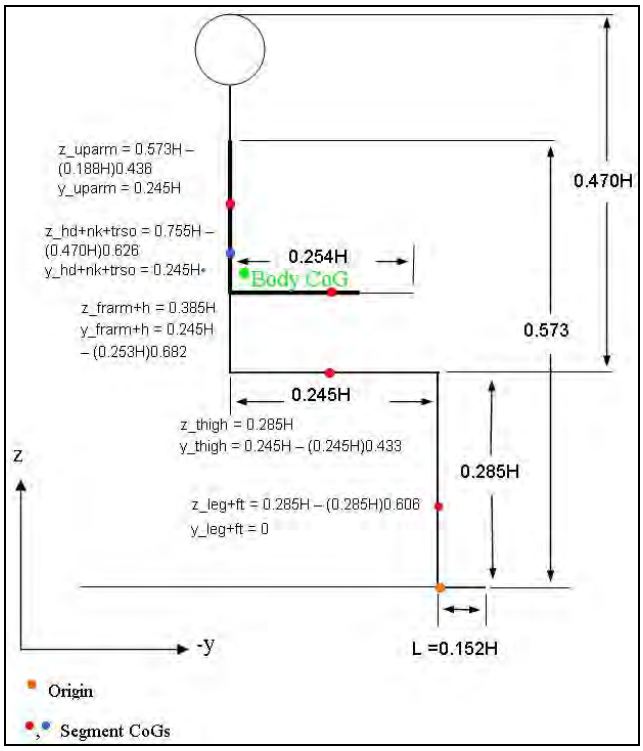

FIGURE 1.6: COG OF SEATED HUMAN BODY

1.4.3 Parameter Values

The parameter values needed for the model's different sub-systems were indicated by the dynamic equations for the wheelchair system. Unfortunately, the wheelchair came with no pertinent drawings or specification data. The particular model has been out of production for some time resulting in little available material. Therefore, full measurements of the wheelchair such as the CoG location, moments of inertia, as well as others, were performed by the University project team of 2006. Motor parameter measurements were also carried out, but re-tests were performed in due course which included identification of a further necessary parameter [29].

1.4.4 Control Applications

A not fully validated version of the FWD wheelchair system model has been used in feedback control design and development with a gyroscope sensor. The design and development has been undertaken in conjunction with Dynamic Controls Ltd (Christchurch, NZ). However, the primary feedback control design and development was carried out almost exclusively by Dynamic Controls personnel.

The model was incorporated into an emulator that is run on a PC. The Emulator mimics a Dynamic Control's motor controller product running an arbitrary electric powered wheelchair. The details of this analysis can be found in Appendix B.

1.5 EQUIPMENT

Through all iterations of research and development of the project there were several pieces of equipment, both hardware and software, used continuously but in a variety of different ways.

1.5.1 Hardware

The prime piece of hardware used, outside of the wheelchair test-bed, was dSPACE, a real-time control implementation and data acquisition system widely utilised by the automotive industry. The device holds several computer cards including: one main processor board and usually several input/output (I/O) boards. The system is modular. Hence all of the boards can be changed or new ones added. However, there must always be at least one

central processor board of which there are different models available. Figure 1.7 is the front panel of the dSPACE platform. Not shown are the I/O connections located at the rear of the device. Table 1.1 indicates the boards used for the current iteration of the project.

FIGURE 1.7: DSPACE REAL-TIME HARDWARE PLATFORM (WOLM ET AL., 2006)

TABLE 1.1: DSPACE MODULAR SET-UP FOR PROJECT

Board	Type	Note
ds1005	Main Processor	PowerPC based technology, 1GHz operating speed
ds4201s	IO	Serial interface (RS232) board
ds814	Link	Provides fibre optic cable link to host PC

1.5.2 Software

Mathwork's Simulink with the Real-Time Workshop (RTW) toolbox and Real-Time Interface (RTI) is the dedicated software of choice for modelling for the real-time dSPACE platform. All means of sending control signals and receiving sensor data from the test-bed

are first modelled with standard Simulink blocks in conjunction with the particular dSPACE I/O blocks from the RTI. Models are then 'built' and compiled into C automatically by the RTW, with a simple command in Simulink, and then transferred to the dSPACE platform, where they operate independently from Simulink.

To access the parts of the Simulink model in dSPACE, a graphical user interface (GUI) known as ControlDesk, which is the software portion of the dSPACE package, is used. ControlDesk sits on the host PC, but communicates continuously with the loaded model on the dSPACE platform in real-time. Various control panels can be built in the GUI and saved and 'paired' with unique Simulink models, as desired. Each control panel can be created with various dials, buttons, sliders, etc. for control and graphs, numerical boxes, etc. for observation of measurements during a real-time test. In Figure 1.8, the Simulink model used in the project to transmit control signals and receive data can be seen. Figure 1.9 indicates the ControlDesk GUI paired to the model.

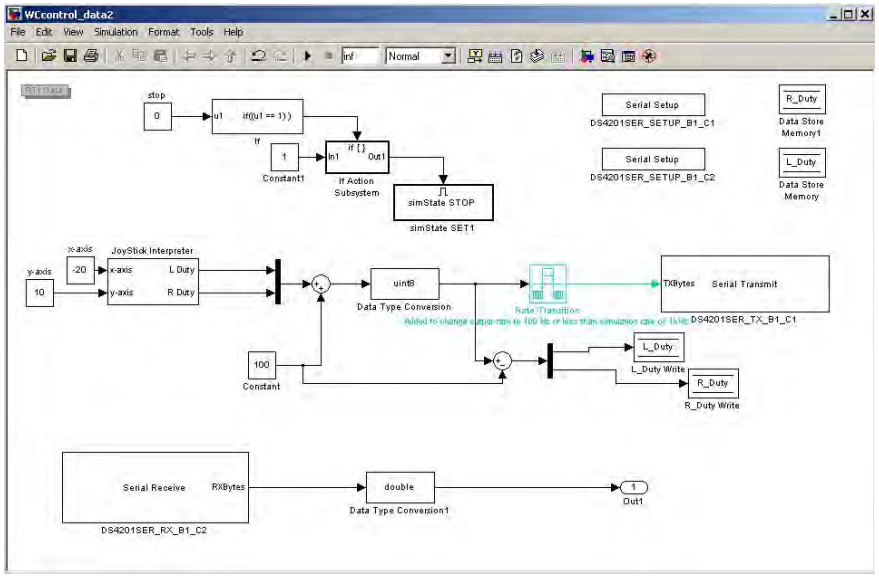

FIGURE 1.8: SIMULINK MODEL TO SEND CONTROL SIGNALS AND RECEIVE DATA TO BE LOADED ONTO THE DSPACE PLATFORM

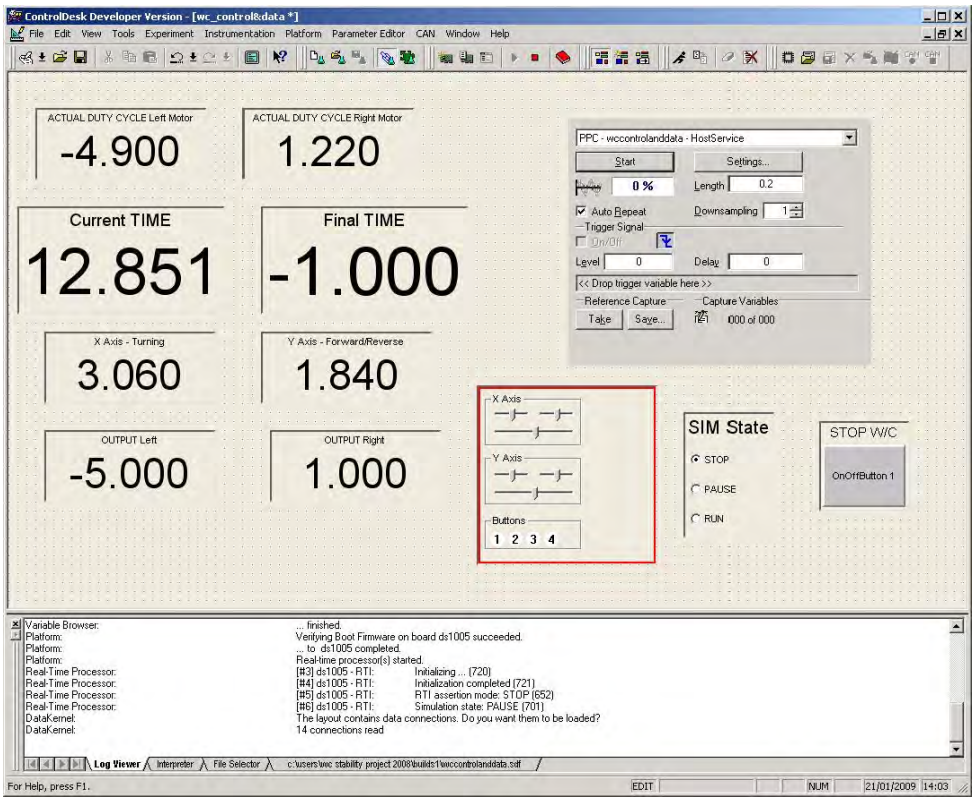

FIGURE 1.9: CONTROLDESK GUI PAIRED WITH THE LOADED SIMULINK MODEL FOUND IN FIGURE 1.8

The box in the upper right quadrant of the GUI is used to capture variables within the model. The system can thus save any type of data that can be found in the model. Hence, all data was saved into a .csv or comma separated file format, which was then opened in Matlab, the basis of the Simulink package, to be parsed and decoded by purpose written M-files.

1.6 PREFACE

Chapter 2 presents a description of the FWD wheelchair test-bed and the control and data collection schemes. It reports the previous developments of a similar test-bed and its

systems and the successful advances made during the current research. Chapter 3 explains the live testing procedures and the methods used to process all recorded data. An analysis of the data is also performed that also includes a discussion of 2 postulated over-steer theories. The details of a Simulink FWD wheelchair model and its mathematics, inherited from previous research are presented in Chapter 4. Some minor changes to the model due to the analysis reported in Chapter 3 and initial inspection during the current research are also covered. The methods of measurement and calculation, and the values of parameters identified from the FWD wheelchair model equations reported in Chapter 4 are described in Chapter 5. All of the parameter values were obtained from prior research [28]. However, Chapter 5 also explains which of these parameters were retained, reconfirmed or changed during the current research. In Chapter 6, the full results of the inherited model are presented in two parts, the PM DC drive motor model and the entire dynamic system model. The testing process for validation is explained first, followed by the presentation of the results and finishing with a full analysis of each model part respectively. The analyses also cover any changes made to the model and the results of those changes. A brief summary of previous discussions is presented in Chapter 8, along with the primary conclusions of this research. Finally, suggestions for future developments leading from this research are discussed in Chapter 9.

2

WHEELCHAIR TEST-BED

2.1 TEST-BED DESCRIPTION

The Permobil Chairman EP FWD wheelchair is a 4 wheeled mobility vehicle with 2 larger diameter rubber compound pneumatic tire driving wheels that are fixed to the chassis at the front, and 2 smaller solid foam caster wheels at the rear. In this report, a caster wheel is considered to be a wheel that can swivel relatively freely (except for a small friction load) on the end of a fixed articulated arm. Table 2.1 shows some basic dimensions of the test-bed and Figures 2.1 and 2.2 provide a view of the Permobil Chairman test-bed from different planes.

TABLE 2.1: BASIC PERMOBIL CHAIRMAN WHEELCHAIR DIMENSIONS

Item	Dimension (m)	Note
Wheel Base	.535	Measured from the contact point with ground of a front wheel to the contact point of the rear wheel on the same side, i.e. from hub to hub
Width	.560	Measured distance between the contact points with ground of the 2 front wheels, i.e. hub to hub
Drive Wheel Diameter	.3414	Measured
Caster Wheel Diameter	.1912	Measured

FIGURE 2.1: FRONT VIEW OF PERMOBIL CHAIRMAN TEST-BED

FIGURE 2.2: SIDE VIEW OF PERMOBIL CHAIRMAN TEST-BED

The test-bed chassis shown in Figure 2.3 is considered to be a single rigid body of solid steel construction for testing purposes. The independent suspension of each articulated arm holding the rear caster wheels was fixed. However, there were certain items that could still be adjusted, such as the seating height and the angle and extension of the footrest within a given range.

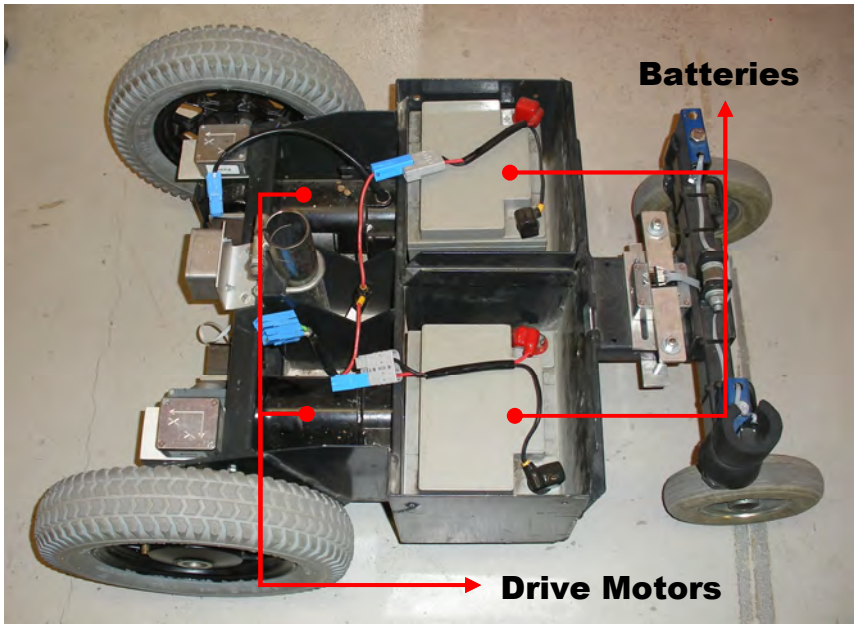

FIGURE 2.3: TEST-BED CHASSIS CONTAINING BATTERIES AND PM DC DRIVE MOTORS

The power for the Chairman is supplied by two 12 Volt (V) 73 Amp-hour batteries that are connected in series to provide 24V. This voltage is supplied, typically via an electronic joystick control, to 2 separate permanent magnet, direct current (PM DC) motors that provide all the driving force. The PM DC motors are connected directly to the driving wheels via a gear box that changes rotation in one plane to rotation in a perpendicular plane. Figure 2.4 shows a close-up of the gear box and one of its attachment points to the

chassis with the black coloured round of the drive motor on the other end. Figure 2.5 indicates a typical EP wheelchair gear box internal layout, as found on the Chairman test-bed.

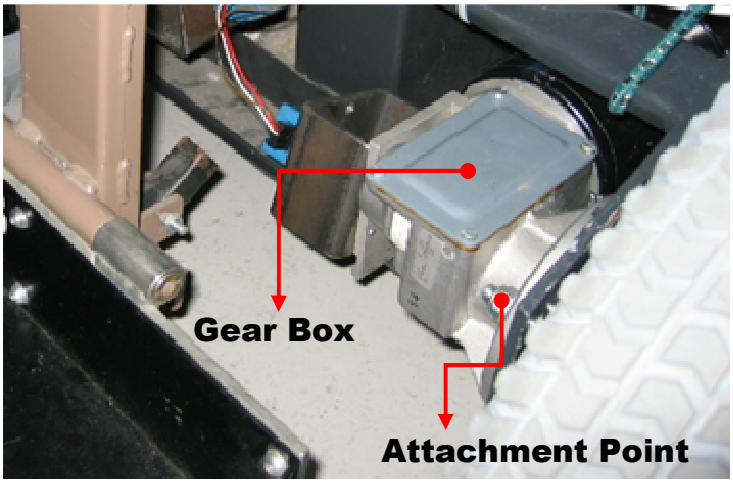

FIGURE 2.4: GEAR BOX AND PM DC DRIVE MOTOR SHOWING A SINGLE ATTACHMENT POINT TO THE CHASSIS
(LESLIE ET AL., 2005)

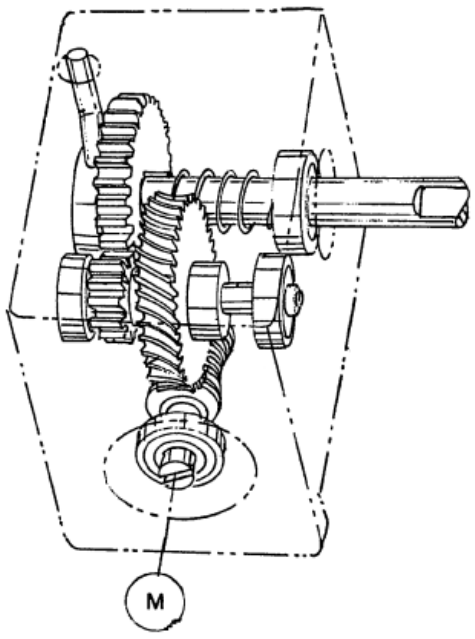

FIGURE 2.5: TYPICAL EP WHEELCHAIR GEAR BOX WITH M INDICATING WHERE A DRIVE MOTOR IS ATTACHED

Steering is provided by using a 'differential drive', as in the case of all main wheelchair drive types. In this design, one drive wheel turns at a different rate than the other, and/or in the opposite direction of rotation, to provide differential tractions and thus turn the wheelchair via direct torque in response to joystick inputs. Thus, the greater the difference in the rates of revolution of the 2 drive wheels, the greater the rate of turn.

The Permobil Chairman test-bed is considered to have a maximum forward velocity of 8 to 10 kmh or 2.22 to 2.78 m/s. These rates hold on a level surface and include the mass of a passenger. Hence the maximum potential turning rate is 9.93 rads/second.

2.2 WIRELESS COMMUNICATION

Bluetooth protocol wireless products were chosen for communication, as they were found to have adequate range of up to 30 metres through reinforced concrete and interference free signal transmission. There is anecdotal evidence that the communication range of these devices can be up to 5 km for line-of-sight. All chosen Bluetooth devices communicated serially, as opposed to in parallel, and followed, or could be easily adjusted to follow, the RS232 serial communication protocol.

Two different brands were used: Brainbox and Sparkfun. Both brands provide compact devices that are similar in size to USB flash drives. The initial attraction for both Bluetooth devices is that any two of the devices could be paired, where they would only communicate with each other and no other Bluetooth devices in range. This characteristic thus ensured

no transmitted data corruption or loss due to interference between or across other devices of different types/brands.

The Brainbox BL-819 devices in Figure 2.6 proved to be highly unreliable in operational lifespan. Of the five ordered for the 2005 project team, one did not function on arrival and one failed during use. The following 3 devices failed during the current research. The replacement Sparkfun devices in Figures 2.7 and 2.8 have proven to be very reliable in all aspects and are still in operation.

However, one of the models of Sparkfun devices, shown in Figure 2.7, required a minor electronic circuit to shift the voltage levels of the signals. The particular Sparkfun model only puts out signals at TTL (transistor to transistor logic) voltage levels, which are different to those as outlined in the RS232 protocol. The PCB level shifter circuit was based on a Texas Instruments MAX232 chip, which is designed to perform these voltage level shifting operations in particular.

FIGURE 2.6: BRAINBOX BL-819 BLUETOOTH WIRELESS RS232 DONGLE (WWW.BRAINBOXES.COM)

FIGURE 2.7: SPARKFUN BLUESMIRF RP-SMA BLUETOOTH WIRELESS DONGLE (WWW.SPARKFUN.COM)

FIGURE 2.8: SPARKFUN FIREFLY BLUETOOTH WIRELESS RS232 DONGLE (WWW.SPARKFUN.COM)

There were two separate channels of communication created. The first is for control and the second for data acquisition. The separate communication avoids signal scheduling, which is complex to institute, creating a simpler design. The separate command channel was also safer because it avoided software induced interrupts of control signals, which if poorly timed, could create instability under control.

The communication set-up consisted of 1 base hub shown in Figure 2.9 and 2 separate Bluetooth dongles on the test-bed as shown in Figure 2.10. The hub handled all transmission to and from dSPACE where all the real-time computations took place. One of the test-bed dongles received control signals while the other transmitted data, each communicating with their respective paired partner at the base hub. Figure 2.9 also shows the baud rate or the serial data transmission rate, which is the number of characters transmitted per second, printed on the hub cover. The rate is set to 57.6 kilobaud, 8 bits per data package, no parity bit and 1 stop bit. Parity is a means for error checking and the stop bit signals the end of each data package.

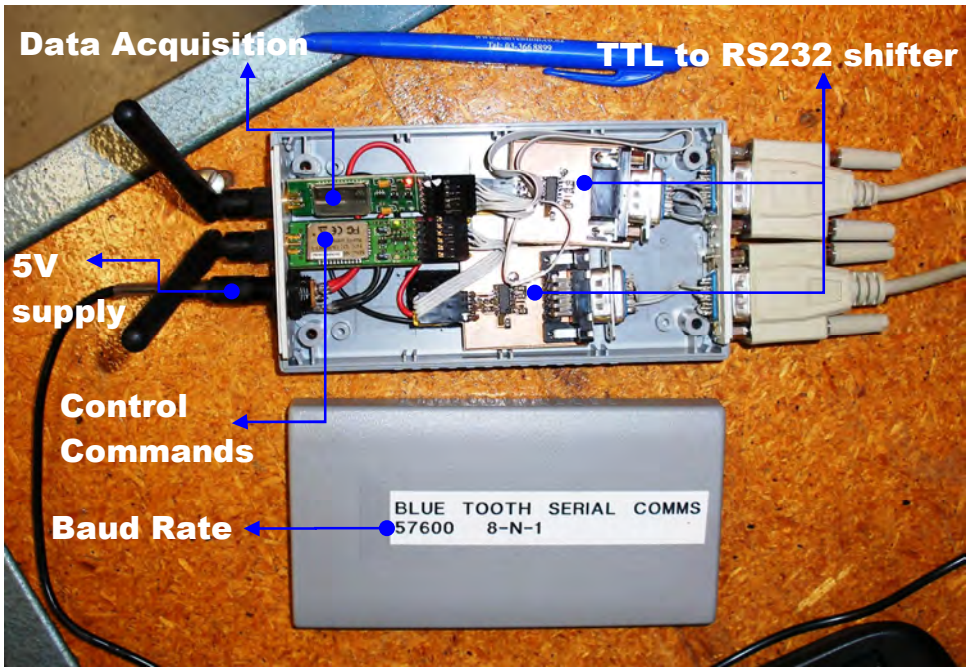

FIGURE 2.9: BASE COMMUNICATION HUB WITH 2 CHANNELS

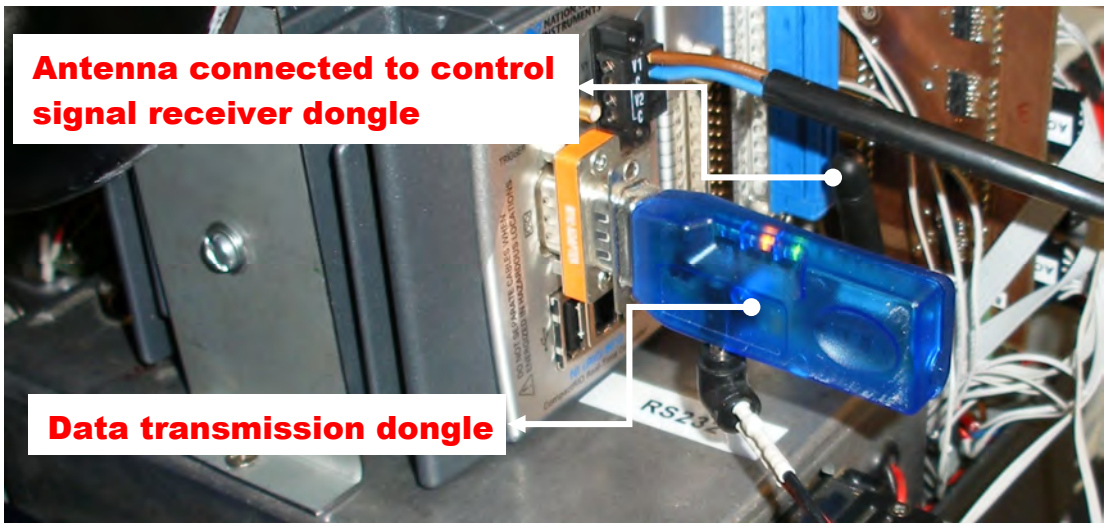

FIGURE 2.10: SEPARATE TEST-BED BLUETOOTH COMMUNICATION DONGLES

2.3 TEST-BED CONTROL DEVELOPMENT

2.3.1 Initial Scheme

In 2005, the project team used an analogue controller provided by Dynamic Controls, as shown in Figure 2.11. Test patterns were driven by a team member using the controller in their version of the Permobil Chairman test-bed. During the tests, the actual internal voltages that were provided by the analogue controller to its motor control hardware were recorded. The voltages were needed to 'play-back' inputs to the wheelchair controller for automated repeatability tests.

However voltage levels from this controller were a problem for the dSPACE platform, which required a specified isolation device to be created. Furthermore, extensive filtering

was required of the recorded voltages, due to the noise in the cable tether over which they were transmitted. The filtering requirement was discovered after playback tests continually tripped breakers within the controller, due to voltage spikes caused by this signal.

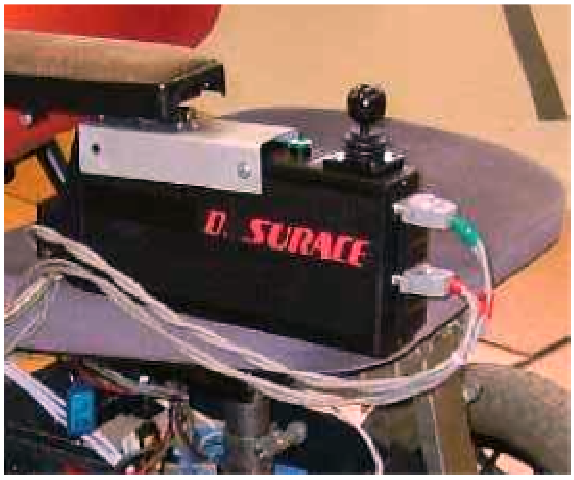

FIGURE 2.11: ANALOGUE JOYSTICK CONTROLLER FROM DYNAMIC CONTROLS (LESLIE ET AL., 2005)

2.3.2 Subsequent Changes

The new project team in 2006 sought to remove the cable tether through wireless control. Several standard RS232 Bluetooth radio transmission serial devices were used and since the serial data is digital, it required a replacement of the analogue controller. To this end, Dynamic Controls provided their Shark Power Module (SPM) wheelchair controller. Figure 2.12 shows this SPM power module into which can be plugged a remote joystick, separate drive motors, battery power, etc.

While the concept was simple, implementation proved to be difficult as the SharkBus protocol, which determines communications within the device, is on a fixed 20 ms 'loop'. This preset, fixed loop timing caused indeterminate delays in communication with the

Bluetooth wireless devices resulting in the loss of transmitted control data. Thus, direct wireless communication between dSPACE and the SPM was not possible. In addition, to communicate with the SPM without a connectable joystick required the use of an Application Programming Interface (API). The API provided for this purpose was a dynamic linked library (.dll) that followed the SharkBus protocol and was run in Matlab.

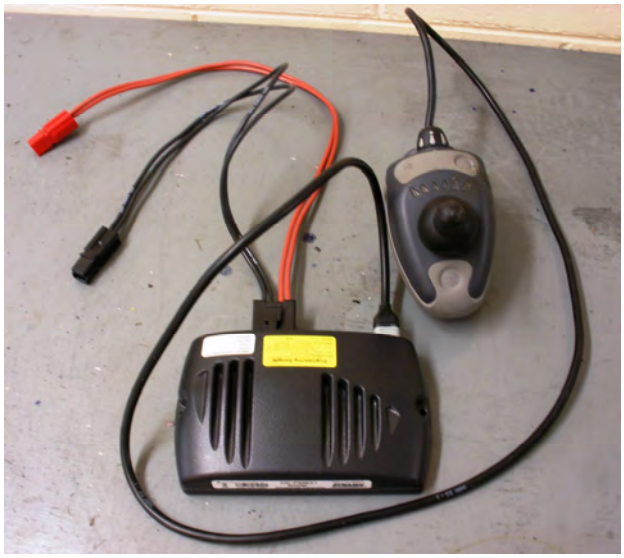

FIGURE 2.12: SHARK POWER MODULE WITH ATTACHED JOYSTICK AND CABLES FOR BATTERY ATTACHMENT

Wireless control was thus achieved, although it was unreliable and needed a complicated equipment set-up. A simple control model was first created in Simulink to be loaded and run from the dSPACE platform. Control signals then went from dSPACE wirelessly to a PC located on the test-bed, which acted as a buffer and was operating the Matlab software. The signals then passed to the SPM via the .dll which was being called by M-files in Matlab. Figure 2.13 shows the equipment overload that was required for this approach, as mounted upon the test-bed. Figure 2.14 illustrates the data path required to achieve control.

FIGURE 2.13: WHEELCHAIR TEST-BED EQUIPMENT OVERLOAD WITH PC 'BUFFER' (WOLM ET AL., 2006)

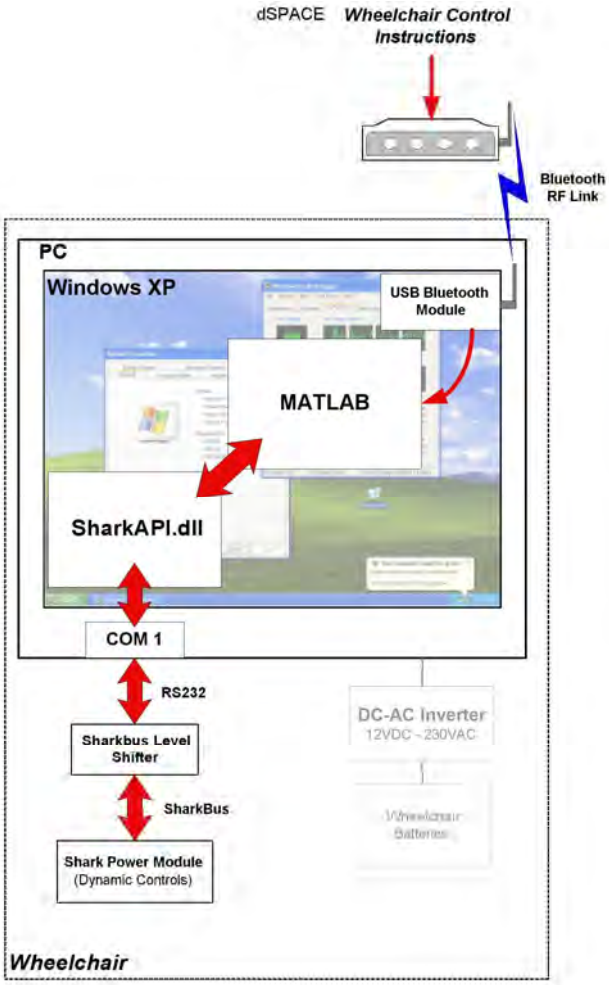

FIGURE 2.14: REQUIRED DATA PATH FOR CONTROL INSTRUCTIONS (WOLM ET AL., 2006)

2.3.3 Current Form

In light of the previous wireless control attempt, it became obvious that a motor controller module without any restricting protocol was necessary. To that end, after consultation with a University of Canterbury (UoC) technical officer, a RoboteQ AX2550 motor controller (Figure 2.15) was chosen. The RoboteQ controller provides independent control for 2 separate motors, allowing both forward and reverse directions, with a maximum supply current of 140 amperes, and operation in the 12 to 40V range. The compactness of the AX2550, 228.5 mm L x 140 mm W x 40 mm H, was also an advantage.

FIGURE 2.15: ROBOTEQ AX2550 DIGITAL DC MOTOR CONTROLLER (WWW.ROBOTEQ.COM)

Another strength of the AX2550 was the ability to command the controller module using the RS232 standard serial communication protocol. This was important because all communication in the transmission chain was based on RS232. However, initial control attempts, through direct communication with the controller, were not successful. Unfortunately, the RS232 protocol does not define a set word length, which is, the number of bits in each data packet that are communicated. The RoboteQ AX2550 uses a 7 bit word

length, whereas many manufacturers, particularly of wireless communication devices, have (unofficially) standardised to an 8 bit word length. It was not possible to change the data package size of the Bluetooth devices (8 bits) and would have necessitated a complex software solution involving bit shifting.

The RoboteQ AX2550 also has 2 further methods of command: analogue voltage input, and radio-control (R/C) pulse width modulation (PWM) input. It was the 2 further methods of command that justified the selection of the AX2550 and also provided a solution to the control issue. Control was therefore established via the PWM input channel. However, a further device, best described as an "R/C PWM input interpreter" or interpreter circuit, had to be designed and built on a printed circuit board (PCB) to accommodate this approach. This board is shown in Figure 2.16.

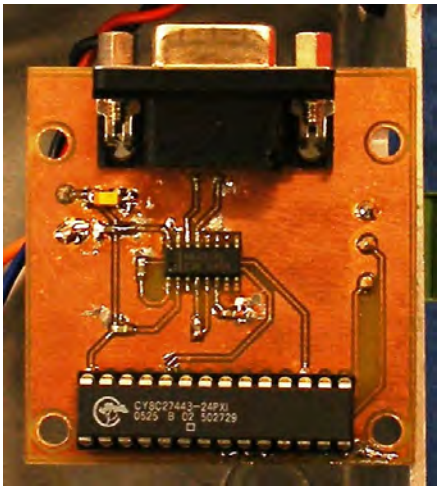

FIGURE 2.16: R/C PWM INPUT INTERPRETER CIRCUIT PCB, 50.6 MM W X 51.6 MM L X 15 MM MAX.H

PWM is a method of modulating a square wave, whose average value is proportional to the pulse width. Figure 2.17 shows this modulation schematically. For example, if only 6 V is

required from a 100 Hz frequency PWM signal with a 12 V input, then the pulse width, time that the signal is on, should be 50% yielding an average of 6 V from the 12 V input. The time that the signal is on (pulse duration or the pulse width) when divided by the period of the main signal of 100 Hz in this example, is more commonly known as the duty cycle in the literature. A frequency, f, equates to the period, T, by T = 1/f. The period of 100 Hz signal is 10 ms and the pulse width or duration for this 6 V example is 5 ms, 50% of the period. Hence the duty cycle in the example is .5 (5 ms / 10 ms) or 50%.

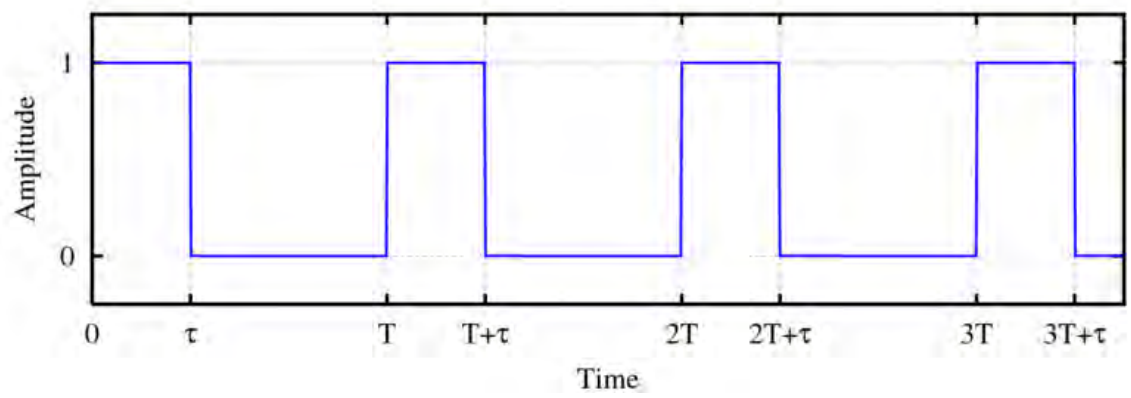

FIGURE 2.17: GENERAL SQUARE WAVE

In traditional R/C controlled vehicles, the communication between the remote controller and the receiver on the vehicle occurs at frequencies in the region of 26 to 76 MHz as mandated by the US Federal Communications Commission. However, R/C control manufacturers are now moving towards using the Bluetooth protocol frequency of 2.4 GHz. The frequency for PWM communication between the receiver and controlled devices on the vehicle is 50 Hz. This value is the same PWM frequency used to communicate with the AX2550 from its receiver. The frequency of 50 Hz is a quasi-standard created by radio control manufacturers, such as Futaba and Hitec, mainly to manipulate servo motors on R/C models.

The 50 Hz communication signal equates to a 20 ms period. In the quasi-industry standard for R/C, to move a servo control to the neutral position (90°) requires a pulse width of 1.5 ms of the 20 ms period. For the minimum (0°) and the maximum (180°) positions, pulse widths of 1 ms and 2 ms are required respectively. The AX2550 controller adheres to this for neutral (stopped), minimum (full reverse), and maximum (full forward) outputs to drive motors. This choice respectively yields pulse widths of 1.5 ms, 1 ms, and 2 ms.

The main means of control for the test-bed is accomplished with a Logitech Wingman Extreme digital joystick as shown in Figure 2.18. A purpose designed control model loaded onto the dSPACE platform reads inputs from the joystick and interprets these motions as duty cycle inputs to the separate drive motors. Appendix C shows the specific details of how joystick position was translated into specific control inputs for the separate motors.

The duty cycle commands are sent via the dSPACE ds4201s serial card to the control channel of the wireless base hub for transmission. However, the ds4201s only allows data in the form of 8 bit unsigned integers. Therefore, to include negative, or reverse, duty cycles, it was necessary to add 100 to all duty cycle commands transmitted and subtract 100 on the receiving side. It was necessary to change duty cycles, such as 100% or -48% sent from dSPACE to meet the PWM input requirements for the AX2550. The interpreter circuit accomplishes both the 100 value subtraction and the changes to the input duty cycle commands.

FIGURE 2.18: LOGITECH WINGMAN EXTREME DIGITAL JOYSTICK CONTROL (WWW.LOGITECH.COM)

The interpreter circuit has a MAX232 for change from TTL to RS232 level. However, the circuit's interpreter itself is a PSoC or Programmable System-on-Chip™ microcontroller from Cypress Semiconductor Corporation, an example of which is shown in Figure 2.19. The PSoC was chosen for its ability to reduce the size of PCBs due to the many functions, which are normally handled via separate devices external to the microcontroller have been internalised in the PSoC. Finally, although the PSoC has a limited number of on-chip hardware, it was perfectly scaled to perform the interpreter function and simple to execute.

Each user has the ability to design their microcontroller set-up using a GUI known as PSoC Designer, also from Cypress Semiconductor Corporation, by choosing from a diverse selection of analogue and digital hardware. PSoC Designer allows one to manipulate the PSoC microcontroller's built in modules within the development environment, which is

shown in Figure 2.20. Selection and movement of the modules is done using "drag and drop" within the GUI.

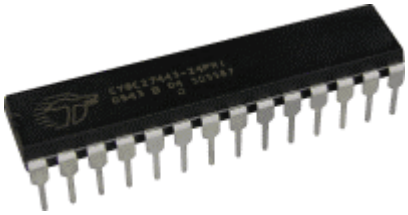

FIGURE 2.19: PSOC MICROCONTROLLER 24 PIN CHIP AS FOUND ON THE INTERPRETER CIRCUIT PCB

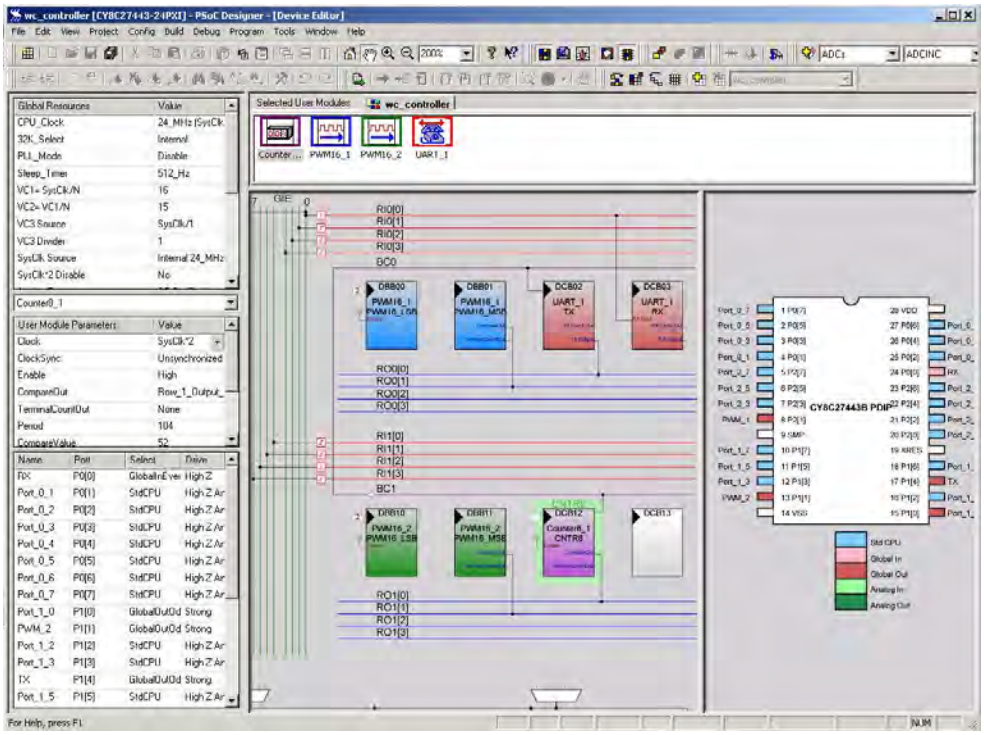

FIGURE 2.20: CYPRESS SEMICONDUCTOR CORPORATION'S PSOC DESIGNER GUI

Referring to Figure 2.20, at the top under "Selected User Modules", are the four selected hardware modules for the interpreter. They are a Counter, used to set the data transmission rate for communications with the base hub, 2 PWM blocks to send out PWM signals to the

RoboteQ AX2550 for each drive motor, and an RS232 protocol UART (Universal Asynchronous Receiver Transmitter) that operates all communications with the base hub.

Directly below the Selected User Modules window is the positioning of those modules, in the digital portion of the PSoC microprocessor, and their internal communication bus connections. The internal communication buses connect to the pins whose physical locations on the chip are indicated by the graphic in the GUI window on the right. Hence, digital modules can be placed anywhere within the digital portion framework. Finally, any pin location for external connection can be selected by choosing the appropriate internal communication buses, and renamed accordingly if desired.

Still referring to Figure 2.20, the windows on the left side of the GUI contain the parameters of each module. The parameters can be seen by clicking on the individual modules, and set as necessary in the left hand windows. Once all of the hardware and parameter selection had been completed, the implementation of the chip was written in both Assembly and C software languages and can be found in Appendix D. Assembly was used to run the framework of the implementation. A C language function was called from the Assembly code to perform mathematical operations particularly as C makes floating point value manipulation much easier when compared to Assembly.

Figure 2.21 shows the entire communication chain for control. Joystick movements are captured by a virtual instrument (VI) in the ControlDesk GUI and sent directly to the Simulink control model that is operating on the dSPACE platform. Changes in model

values that are monitored by VIs in the ControlDesk GUI are updated in real-time by dSPACE. Furthermore, dSPACE sends out the control commands captured by the joystick virtual instrument through its serial card to the control Bluetooth dongle in the Base Communication Hub, also in real-time. The Hub Bluetooth dongle communicates with it slave on the test-bed wheelchair. The test-bed wheelchair Bluetooth dongle passes its signal directly to the R/C PWM Input Interpreter circuit that modifies the transmitted duty cycle demands to the appropriate PWM values required by the Roboteq motor controller. The Roboteq unit then interprets the PWM values to send the appropriate voltage demands to the drive motors.

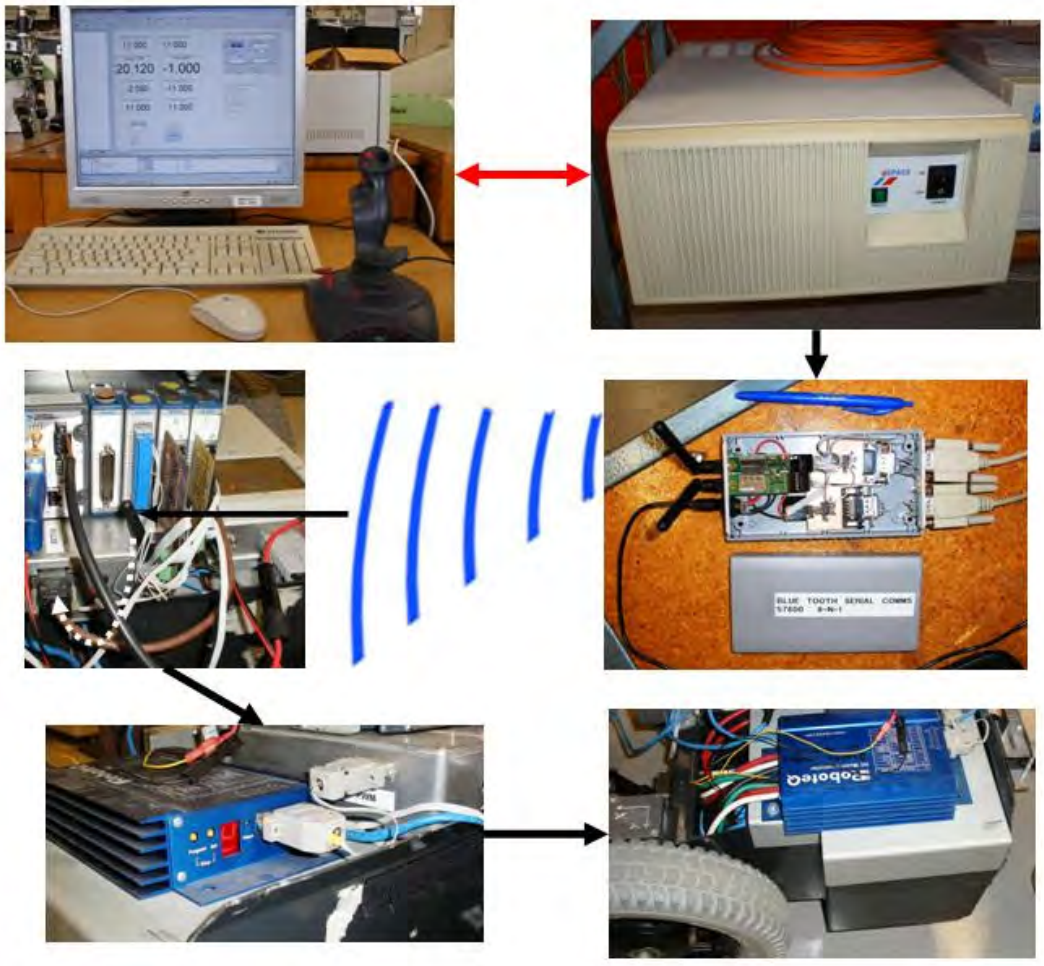

FIGURE 2.21: CONTORL COMMUNICATION CHAIN

2.4 DATA ACQUISITION DEVELOPMENT

2.4.1 Initial Scheme

The same issues experienced with control of the wheelchair test-bed, also affected data acquisition, as well. These issues were largely due to the large signal interference associated with the cable tether used, and the scale of the equipment needed to make the system work. Figure 2.22 shows the excessive amount of equipment required to manage data acquisition in the 2005 incarnation. Once again, filtering was only partially successful at managing the noise and other issues.

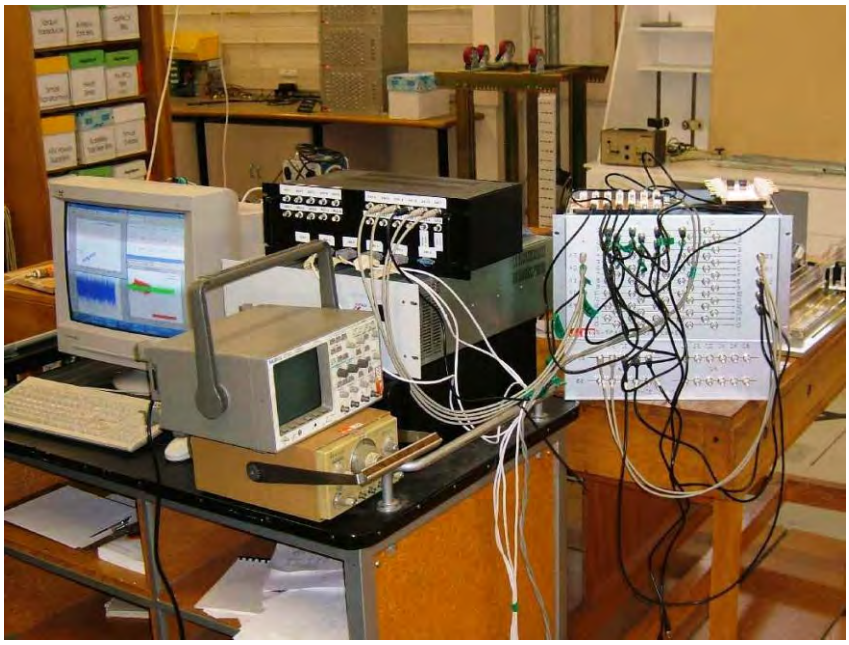

FIGURE 2.22: DATA ACQUISITION WITH TOO MANY CABLES AND EQUIPMENT (LESLIE ET AL., 2005)

2.4.2 Subsequent Changes

With the change to wireless transmission, a new means of collecting the data was necessary from the test-bed sensors. In addition, more data sensors were added overall, even though some others were removed. The sensor set-up is shown in Figure 2.23.

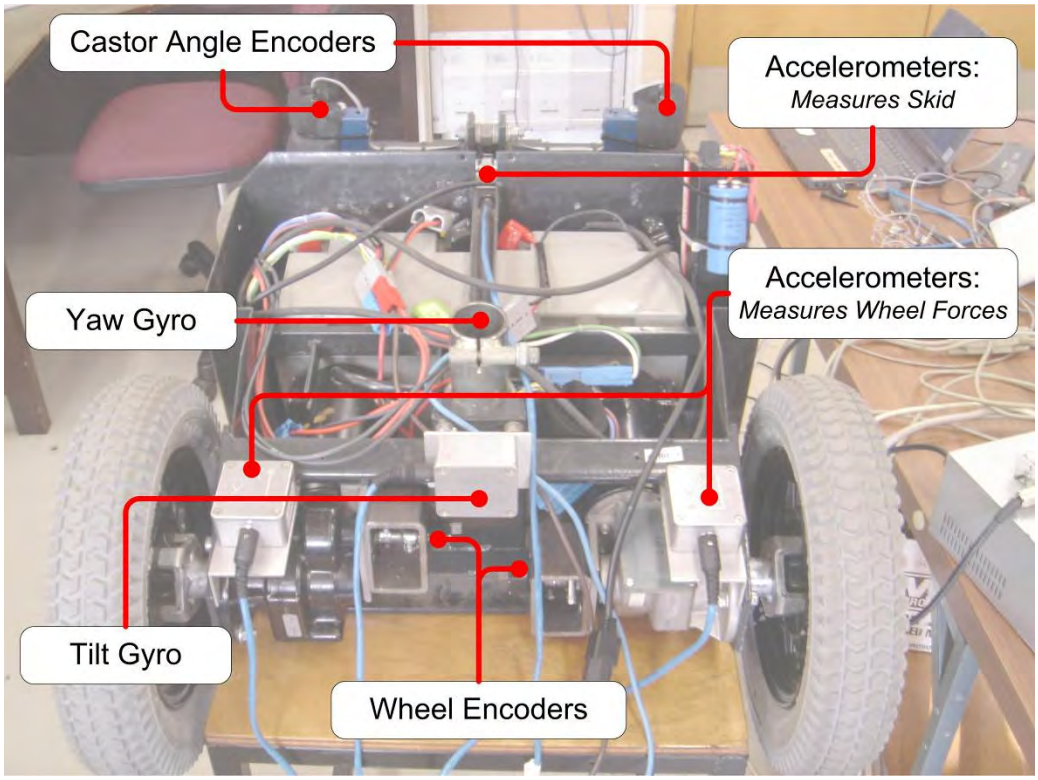

FIGURE 2.23: TEST-BED SENSORS AND THEIR LOCATIONS (WOLM ET AL., 2006)

The technical officer for the team suggested a system where, during each step of transmission, all the data from each sensor was to be sent in a given order at one time. The set data order provided a means of identifying which data matched which sensor when decoding and translating the data at a later point. In addition, all data was to be transmitted asynchronously, meaning that transmission would occur immediately when the system was turned on and not wait for a start signal. Therefore, a means was also needed to identify the

start and end of each set sensor data stream that was transmitted per time step. Table 2.2 indicates the makeup of the 122 byte transmission steam transmitted each time step, where each byte equals a word length of 8 bits. It should be noted that the data stream contains more data places then actual sensors. This difference was designed for the possible future addition of more sensors. In addition, the added data places did not detract from transmission speed or accuracy.

TABLE 2.2: MAKEUP OF TRANSMITTED 122 BYTE DATA STREAM

Order in data stream	Item	N^{o} of bytes	Note
1	Start	1	ASCII value for New Line character
2	Gyroscope data	24	3 x gyros, 8 bytes each
3	Accelerometer data	64	4 x two axis accelerometers, 8 bytes per axis
4	Rotary encoder data	32	4 x encoders, 8 bytes each
5	Stop	1	ASCII value for Carriage Return character

The team's technical officer also designed and built a system in-house, best described as a "data collation device" (DCD) to accomplish the collating and ordering of the various sensor data. It is shown in Figure 2.24. The start and stop bytes were added by the DCD before each data stream was sent for wireless transmission to the base hub. In addition, the DCD provided all levels of voltage supply for the various sensors and Bluetooth dongles, such as both 5 V and 8 V. This task was accomplish by converting the 24 V supply from the test-bed batteries. However, no data collection occurred due to the control problems discussed in section 2.3.2.

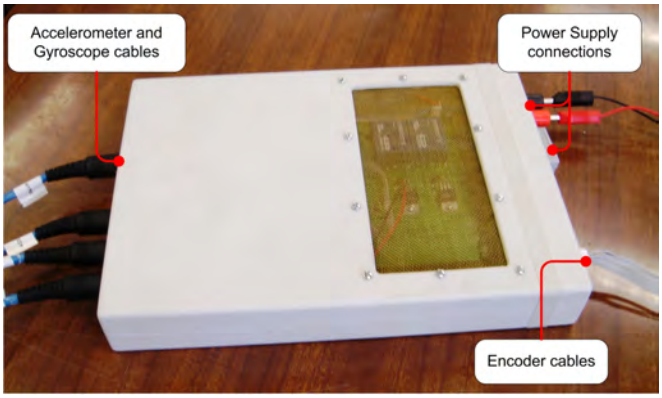

FIGURE 2.24: TEST-BED SENSOR DATA COLLATION AND POWER SUPPLY DEVICE (WOLM ET AL., 2006)

2.4.3 Current Form

The same sensor makeup has been retained. However, the rear left Castor Angle rotary encoder, as seen when viewed in Figure 2.23, failed and replacement of the device was deemed non-critical as collection of castor angle was not a prime aim of the research. Table 2.3 indicates the type, model, and features of sensors used. When data collection attempts were finally made, the DCD did not work properly as no encoder data was sent through due to a technical problem. After a prolonged period of attempts to repair the DCD, the technical officer suggested an off-the-shelf replacement system. However, the DCD was retained for its power conversion and distribution services.

The replacement chosen was a National Instruments CompactRIO shown in Figure 2.25. The CompactRIO is a rugged reconfigurable real-time device that National Instruments describes as “an advanced embedded control and data acquisition system”. It is composed of an embedded real-time controller, which can operate in a stand-alone mode or connected by Ethernet to a host PC. A chassis contains a field programmable gate array (FPGA), the core of the CompactRIO system. Individual I/O modules slot into the chassis. A

programmed FPGA can control fast data acquisition and/or control outputs through direct individual connections to each I/O module. It also provides the connection between I/O and the real-time controller via an internal communications bus. Table 2.4 indicates the type and configuration of the CompactRIO used in this research with the real-time controller on the left.

TABLE 2.3: DATA SENSORS USED IN THE RESEARCH

Sensor	Manufacturer and Model	Features
solid state gyroscope	Analog Devices ADXRS300	• single axis yaw rate response on a single chip • measurement range: ± 300 °/second • sensitivity: 5 milli V/ °/ second • 2000 *g* powered shock operation
solid state accelerometer	Analog Devices ADXL213	• dual axis accelerometer on a single IC chip • measurement range: ± 1.2 *g* • 1 milli *g* resolution at 60 Hz • 3500 *g* shock survival
rotary optical encoder	Bourns Inc. ENCIJ	• 2 channel quadrature output • resolution: 256 Pulses Per Revolution (PPR) • small package size: 21mm x 16mm • maximum rate: 300 rpm

FIGURE 2.25: COMPACTRIO EMBEDDED CONTROL AND DATA ACQUISITION SYSTEM (WWW.NI.COM)

TABLE 2.4: RESEARCH COMPACTRIO MODEL AND CONFIGURATION

Item	Model	Notes
Real-time controller	NI cRIO-9012	• floating point 400 MHz Freescale processor • 128 Mbytes of non-volatile flash memory • 64 Mbytes DRAM • contains 1 x RS232 serial port and 1 x USB port for Ethernet connection
FPGA chassis	4 slot, 1 M	• FPGA contains 1 million gates
Analogue Input module	NI 9205	• 1 x analogue module for gyros • 32 single ended or 16 double ended inputs • 16 bit resolution • 250 KiloSamples/s aggregate sampling rate • ±200 mV, ±1 V, ±5 V, ±10 V programmable input ranges
Digital I/O module	NI 9401	• 2 x digital modules for accelerometers and rotary encoders • 8 channels, bi-directional • 100 ns ultra – high speed digital I/O

Applications known as virtual instruments or VI's were created in LabVIEW, the graphical programming language of choice for the CompactRIO. The LabVIEW project containing the VI's in Figure 2.26 was then loaded onto the CompactRIO platform to operate in stand-alone mode. Appendix E contains the graphical code for both the FPGA and real-time controller. The FPGA controlled all input from the test-bed sensors via the I/O modules, and to the real-time controller. All the data was collated in the real-time controller in the same manner and pattern as listed in Table 2.2, and transmitted wirelessly through the controller's serial port. Supply voltage for the CompactRIO was taken directly from the 24 V test-bed batteries via the DCD, which fell in the required range of 9 to 35 V.

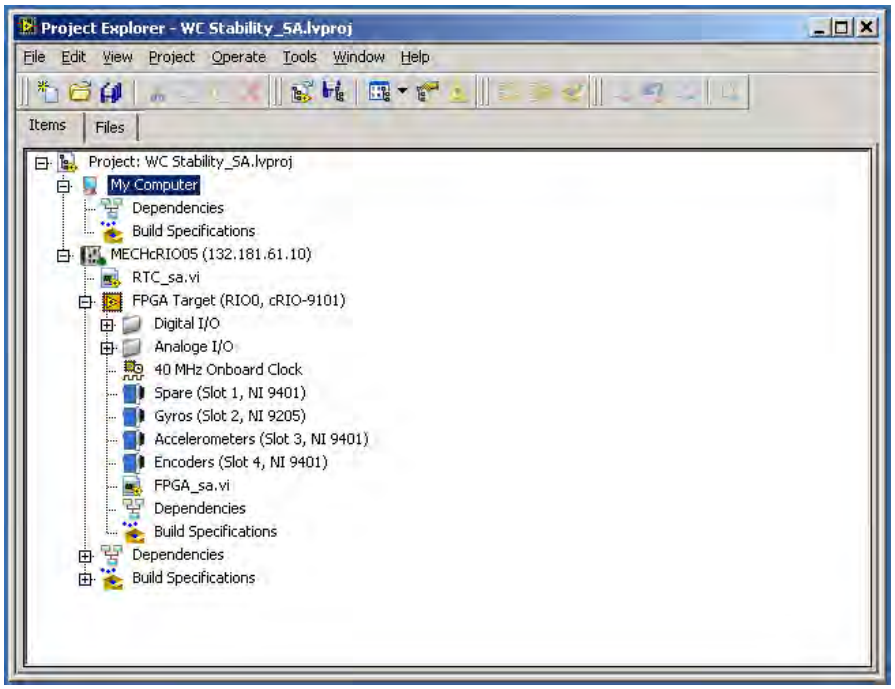

FIGURE 2.26: LABVIEW GUI SHOWING DATA ACQUISITION CONTROL PROJECT THAT WAS LOADED ONTO COMPACTRIO

2.5 SUMMARY

A FWD wheelchair test-bed has been successfully developed. The test-bed contains several different sensors retained from previous research. Wireless control and data collection systems have been implemented to overcome prior prototypes that restricted wheelchair motion and/or had an unreliable wireless control system with a large amount of sensor noise. The test-bed development also required new ancillary devices and/or new methods of operation for the ancillary devices retained from prior prototypes. The overall chapter thus provides a full description of the test-bed and the methods, systems and devices used to control the test-bed and to capture and process data.

3

DATA ANALYSIS

3.1 INTRODUCTION

Dynamic data acquired during live tests of the FWD test-bed were collected for two purposes: 1) to confirm theories that explained the cause of the over-steer phenomena; and 2) to be used to verify the FWD wheelchair model. A series of live dynamic tests were performed to observe and record events under the conditions of the theory. Static tests were also performed to verify hardware processes and operations.

The data from the wheelchair test-bed sensors was recorded in .csv files and needed to be processed through several purpose written Matlab M-files. Individual M-files parsed, converted, decoded, and turned the raw data into representable measurements and graphs for each sensor. A full list of the Matlab M-files and a description of their operations can be found in Appendix F.

3.2 TESTING PROCESS

Dynamic tests were required to understand the nature of over-steer and find the limits of controllability, the point or points at which the wheelchair spun out of control. Any test profile that provided turns would be deemed adequate. However, information obtained via the sponsoring company indicated that a profile described as a 'flat-S' is one of the most difficult manoeuvres to perform while maintaining control in a FWD wheelchair. Figure 3.1 shows a plan view of the ground trace of the flat-S profile defined as a dog leg pattern. An oval ground trace pattern was also driven to obtain data under another set of turning conditions, as shown in Figure 3.2.

The oval pattern was driven in a clockwise direction when viewed from above, with all turns to the right. There were two profiles of each dogleg pattern. The first profile was driven with the initial turn to the left, or anti-clockwise, and the second had the initial turn to the right. Both the oval and dog leg patterns were driven in the forward direction that an occupant faces. In addition, the two profiles of the dog leg pattern were also driven in reverse. Table 3.1 indicates the number of tests performed and under what conditions. It should be noted that all tests were performed on a relatively level surface and loss of control indicates the test-bed spun.

FIGURE 3.1: FLAT-S PROFILE KNOWN AS A DOG LEG PATTERN

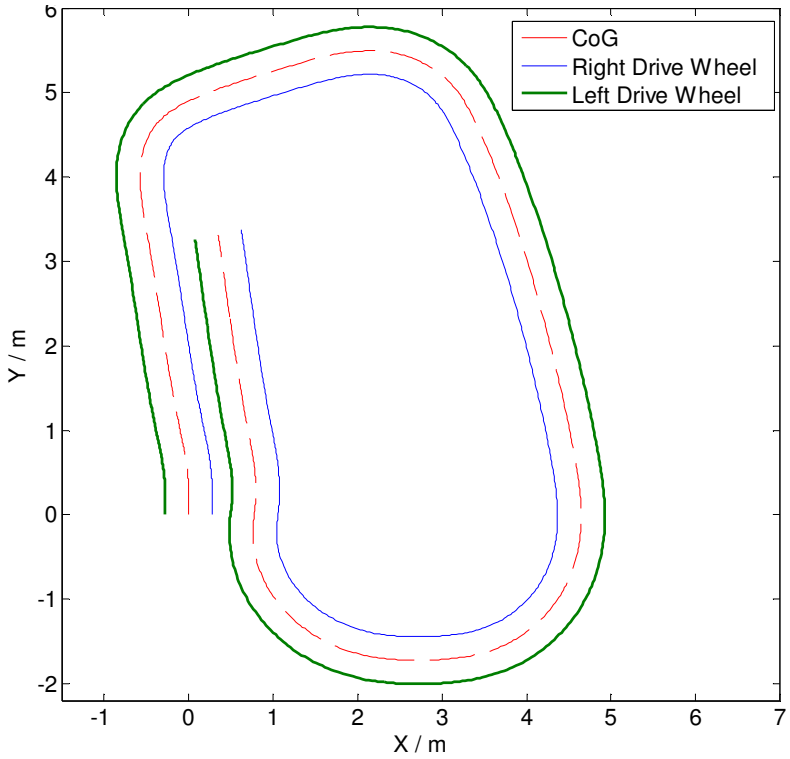

FIGURE 3.2: PLAN VIEW OF OVAL PATTERN WHEELCHAIR TEST DRIVE TRACE

TABLE 3.1: NUMBER OF DYNAMIC TESTS AND CONDITIONS

Pattern	Under Control		Loss of Control	
	Occupied	Unoccupied	Occupied	Unoccupied
Dog Leg	2	4	2	2
Oval	1	1	–	–

Initially, all tests were performed with the Logitech joystick as the sole means of control. However, after an analysis of a preliminary series of tests, it was discovered that there was a miscorrelation, as seen in Figure 3.3 between the control inputs and the data acquisition. In particular, an indeterminate delay would occur from when the first control signal was sent to when data acquisition started even though there was instantaneous motion from the test-bed. This undefined or variable delay was found across all test pattern results. It was discovered that the joystick interpreter shown in Figure 3.4, which was part of the Simulink

control model, was a source of delay. It has not been confirmed whether it was the joystick interpreter itself or the SteeringController virtual instrument found in ControlDesk was the cause. The SteeringController, seen in Figure 3.5, is used by dSPACE to identify any type of game controller attached to the system, such as the Logitech joystick, and capture values from the game controller. The values captured by the SteeringController were fed to the joystick interpreter in the Simulink control model loaded onto the dSPACE platform.

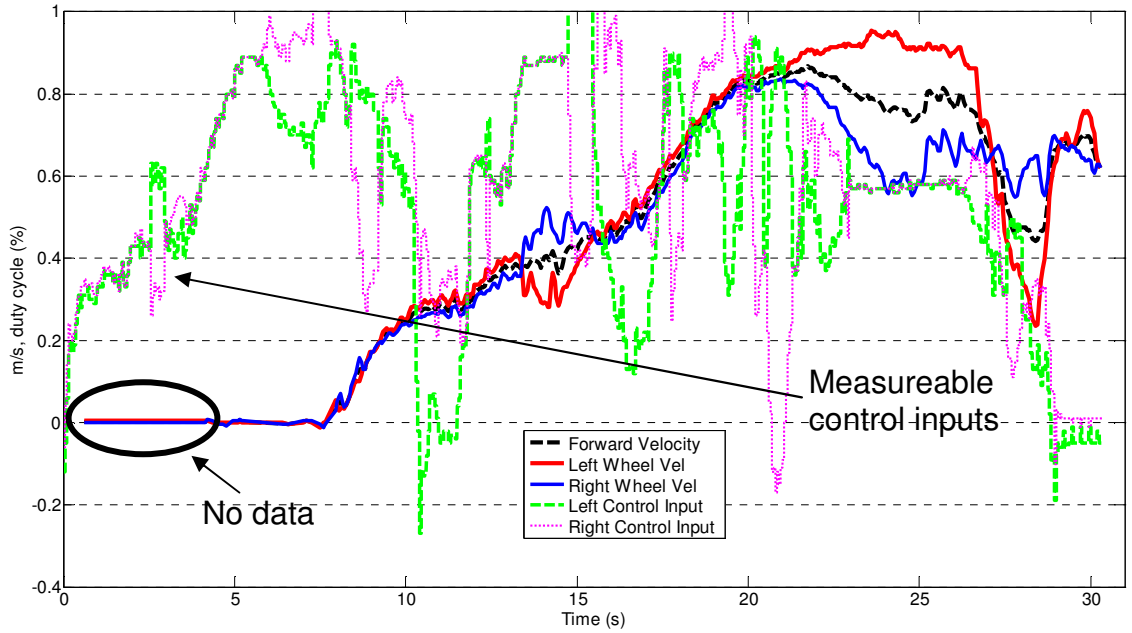

FIGURE 3.3: PROCESSED DATA THAT HAS BEEN CONVERTED TO WHEEL VELOCITIES AND OVERALL WHEELCHAIR VELOCITY WITH AN AREA OF NO DATA, FOLLOWED BY A DELAY IN DATA RESPONSE EVEN WITH CONTROL INPUTS OVER 60%

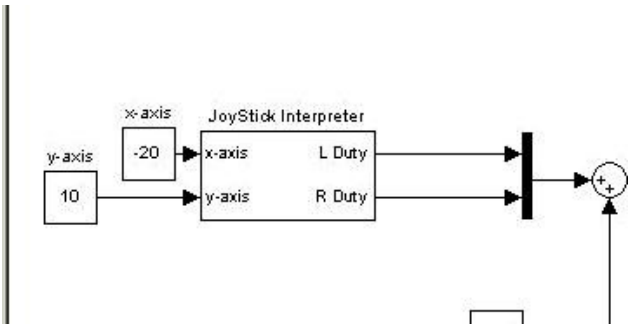

FIGURE 3.4: JOYSTICK INTERPRETER SUB-SYSTEM FOUND IN SIMULINK CONTROL MODEL THAT READS INPUTS FROM THE LOGITECH JOYSTICK VIA CONTROLDESK/DSPACE

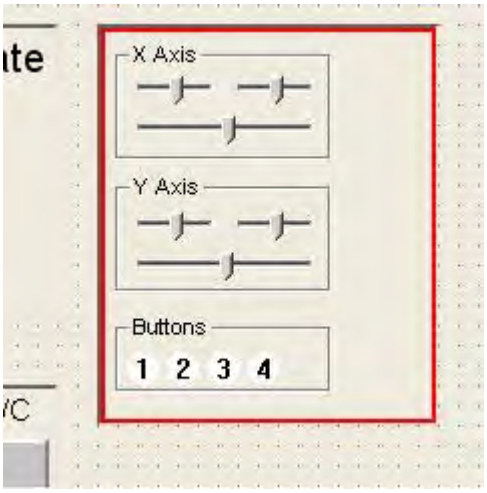

FIGURE 3.5: ISOLATED VIEW OF THE STEERINGCONTROLLER VIRTUAL INSTRUMENT IN CONTROLDESK/DSPACE

The solution was to drive all the test patterns using the Logitech joystick, while recording all the transmitted control inputs through ControlDesk. The control inputs were then processed, saved in a Matlab .mat file format, and used in another control model, which is shown in Figure 3.6.

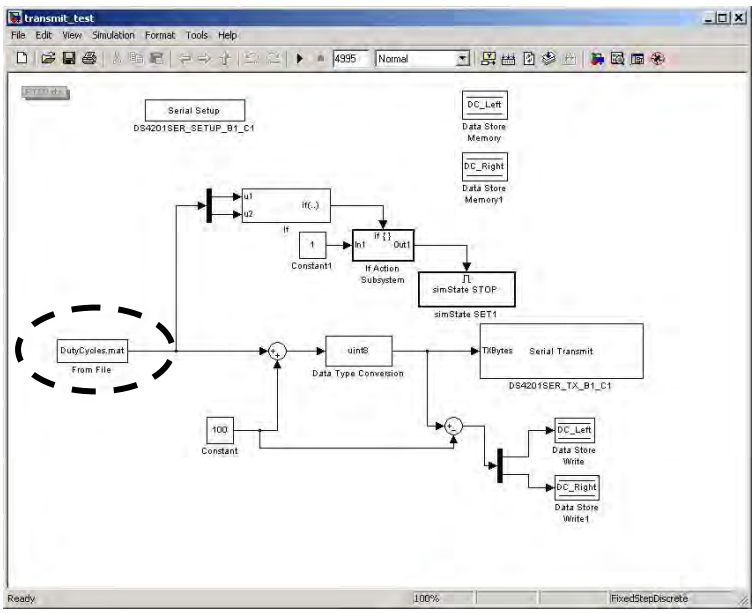

FIGURE 3.6: CONTROL MODEL WITH INPUTS PROVIDED FROM A SAVED .MAT FILE REPLACING THE JOYSTICK INTERPRETER

The .mat file is input via a “From File” Simulink Block, which thus replaces the joystick interpreter block and automates the control process. The result was that the correlation issue between all of the data acquisition and transmitted control inputs was solved for all test patterns in a safe and reliable manner.

3.3 DATA PROCESSING

The raw data went through a preliminary master m-file that was the core of the data processing. In addition, there were several ancillary M-files, divorced from the master m-file, and thus not called as sub-functions. These functions were written to further refine the data and create more plots. The preliminary processing in the master m-file adhered to a set process defined:

1. Open the saved data .csv file in Matlab with master m-file.
2. Call a sub-function from the master m-file to remove ancillary additions to the file, reorder the data, split the data into separate calibration, actual test results, and control input values, and remove start and stop values for each data stream line.
3. Call a sub-function from the master m-file to convert all values to their proper decimal values.
4. Call a sub-function from the master m-file to change the raw data of the separate calibration and valid results into raw values of floating point measurements for gyros and accelerometers, and integer values for the rotary encoders.
5. Call a sub-function from the master m-file to convert the raw floating point measurements and rotary count values into actual roll and yaw rates for the gyros,

acceleration values for the accelerometers, and values from the encoders into angle changes, overall distance changes, and rotational distance and forward distances per time step for each drive wheel.

6. Call the final two sub-functions from the master m-file that plot the trajectory of the test-bed wheelchair during individual tests and reformat the control input values for direct use in the FWD wheelchair model.
7. Save final decoded data from step 5.

Step 1

Processing of saved .csv file started by loading and opening with MumProgram3 that also goes on to perform steps 1-7. Example: MumProgram3('test.csv')

Step 2

When saving the incoming sensor data to .csv files, dSPACE adds additional information to the data file such as time of data collection start, finish, etc that needs to be ignored to access the raw data values. Furthermore, due to a particular design feature in ControlDesk/dSPACE, the order in which the set pattern of each byte of the data stream is sent is saved in a set manner that does not follow the original transmitted order. Each data stream is saved as an individual vector for each time step. Thus, for example, data byte [1, 2] in the data stream is followed by byte [1, 20], [1, 21], [1, 22], … instead of byte [1, 3], [1, 4], … representing the individual sensor data. However, once all issues of dSPACE's particular method of saving data were understood, the .csv file format proved very simple and expedient to parse.

Control inputs that are transmitted to the test-bed during testing are saved through ControlDesk and added to each data stream vector by dSPACE. The control values do not need the same rearrangement as the actual data. However, the control values are separated before removal of the start and stop bytes in the raw data stream can begin.

Before starting dynamic or static tests, data recording is initiated before any actual control inputs are sent for a period of 5 seconds. This early recording is performed to calibrate the output of all the gyro and accelerometer sensors before each test. The calibration values are removed after the control inputs from the .csv data file and saved separately as well. The calibration values are then used to provide the zero when calculating the actual yaw, roll, or acceleration rates.

Step 3

All data was sent back in ASCII (American Standard Code for Information Interchange) format. This format was chosen for debugging or error checking purposes as the data could be viewed in a simple terminal emulator, such as HyperTerminal. In this manner, a readable message could be sent to test the function of the sensor data collation device on the test-bed or wireless dongles when needed.

Each byte of the sensor data is a decimal integer value that represents an ASCII character, which can be either a letter in the English alphabet, a number with values from 0 to 9, or other defined symbol. The ASCII characters are treated as hexadecimal values, which is a base 16 number system commonly used in computing. For instance, the decimal integer 65

in ASCII terminology represents the English letter 'A', which is equal to 10 in hexadecimal. Once all data was converted to its proper hexadecimal format, it was returned the master m-file for further processing.

Step 4

Each sensor's various measurements are encoded in an 8 byte 'word'. With each byte in a word now in hexadecimal form, a set procedure for conversion was to change the 8 hexadecimal numbers to a single decimal value. The hexadecimal values of the gyro and encoder data were converted directly to a decimal with the procedure shown by Equation 3.1. Note that *a* – *h* represent possible hexadecimal values in a gyro or encoder 8 byte data word with *a* as the first byte going from the left to right with *h* naturally being eighth byte. All encoder data was left in its single decimal form. Gyro decimal values were changed into binary for conversion to floating point values.

$$\boldsymbol{a}\times 16^{7}+\boldsymbol{b}\times 16^{6}+\boldsymbol{c}\times 16^{5}+\boldsymbol{d}\times 16^{4}+\boldsymbol{e}\times 16^{3}+\boldsymbol{f}\times 16^{2}+\boldsymbol{g}\times 16^{1}+\boldsymbol{h}\times 16^{0} \qquad (3.1)$$

The accelerometer data was converted differently. Each hexadecimal value in the 8 byte word was treated as a binary nibble or half byte in its more traditional sense. Thus, each hexadecimal value representing one byte of the 8 byte word was converted directly into its equivalent 4 bit binary value. Each 4 bit binary nibble was concatenated together in order, from left to right. This concatenation formed a 32 bit binary value.

All conversions of decimal values to binary were accomplished using a built-in Matlab function dec2bin. However, a function for conversion from binary to floating point values

had to be created. The coded algorithm for the binary conversion function was based on the IEEE (Institute of Electrical and Electronics Engineers) Standard for Binary Floating-Point Arithmetic 754. The binary representation of the 8 byte data words consists of 32 bits, which is defined as "single precision" under IEEE 754. Table 3.2 shows the IEEE process the written algorithm used to convert to floating point for base 2 numbers with the 32 bit binary value example of 00111111000001101001001101001101. Note that in the example the bits are read from left to right. However, by convention, binary values are normally read right to left.

TABLE 3.2: CONVERTING A BINARY VALUE TO FLOATING POINT

sign	**exponent**	**Significand or Mantissa**
1st bit	bits 2 - 9	bits 10 – 32
0	01111110←	→00001101001001101001101
formula: $(-1)^{sign} \times 2^{exponent-127} \times (1 + \textbf{\textit{fraction}})$		
Note: Mantissa is used to calculate fraction		
Formula step	**Result**	**Note**
$(-1)^0$	1	Value is positive
$2^{126 - 127}$	0.5	Conversion of binary exponent to decimal value = $2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1$ (bit position starting from the right (←) where value is "1", starting count from 0)
1 + 0.051370	1.051370	fraction = $1/2^5 + 1/2^6 + 1/2^8 + 1/2^{11} + 1/2^{14} + 1/2^{15} + 1/2^{17} + 1/2^{20} + 1/2^{21} + 1/2^{23}$ (bit position starting from the left (→)where value is "1", starting count from 1)
final result: 1 x 0.5 x 1.051370 = 0.525685		

Step 5

The floating point values for the gyros and accelerometers were converted to their actual rates using formula's provided by the device manufacturer. Both types of devices were manufactured by Analog Devices Inc., as noted in Table 2.3. According to the respective device's data sheets the formulae are as follows:

- Gyro: (measured output / sensitivity),

  where sensitivity = 5 milliV/°/second. The measured output, which is an analogue voltage, is taken directly from the sensor.

- Accelerometer: (measured output – Zero g bias) / sensitivity ,

  where the Zero g bias was calculated from calibration measurements (1^{st} 5 seconds of each test run) and sensitivity = 0.3 g. The actual output from the sensor is a digital PWM signal. The CompactRIO is used to calculate the pulse width and period of the PWM signal from the sensor and thus the duty cycle. The duty cycle is the actual measured output in the formula. Hence, + 100% duty cycle equals + 1.2 *g*'s and - 100% duty cycle, equals -1.2 *g*'s

It should be noted that the gyro data was also corrected for zero g bias from the calibrated data to filter any noise.

The encoder data was broken into several streams of results that were saved separately. The streams were the encoder counts, the distance travelled, and the radians rotated for each drive wheel. The distance travelled was calculated with:

(encoder count change each time step / number of encoder counts for one revolution of a front drive wheel) * front drive wheel circumference. (3.2)

The radians rotated were calculated with:

(encoder count change each time step / number of encoder counts for one revolution of a front drive wheel) * 2 pi. (3.3)

Step 6

The trajectory of the wheelchair test run is plotted in plan view XY plane showing the trace of each drive wheel and CoG. The position of each drive wheel is plotted for each time step from a calculation of a position vector derived from the encoder counts. The control input data is saved in column arrays. However, the Simulink FWD wheelchair model prefers inputs to be in row arrays. Thus, they are converted from [: , 3] to [3 , :].

3.4 ANALYSIS

3.4.1 Introduction

The data was analysed using gyroscope and encoder sensor measurements. The use of only two of the types of sensors available was done for simplicity. Gyroscope data required very little post processing, mainly a change from °/s to radians/second and integration of

rate data to obtain angles. The encoder data, to be changed to linear and rotational velocities, required more post processing that was trivial to institute. Accelerometer data was primarily gathered for model verification. The accelerometers of primary importance on the test-bed were placed at the central axis of the drive wheels to calculate forces experienced by them.

3.4.2 Initial Over-Steer Theory

An initial theory proposed that the inside wheel would lift each time an occupied FWD wheelchair was in a turn on a level surface. The wheel lift theory postulated that when a person sat in the wheelchair, it would raise the CoG of the wheelchair/occupier system. The raising of the CoG allowed a greater tilting moment to occur in turns and the greater the velocity, the higher the tilting moment arising, resulting in greater wheel lift.

The force acting on the CoG that caused the tilting moment is created by an opposition to the change in the direction momentum of the wheelchair when turning, as seen in Figure 3.7. The over-steer would occur because of an extreme differential driving force between the drive wheels similar to the loss of traction of one drive wheel. Furthermore, over-steer was created or exacerbated mainly due to this lifting.

Another theory, the proximal theory, was proposed at the same time that also had an effect on the over-steer phenomena. The proximal theory postulated that since the CoG lay behind line of corning force, as the distance of the CoG behind the increased, so to would over-steer instability in accordance with Collins and Kauzlarich [3].

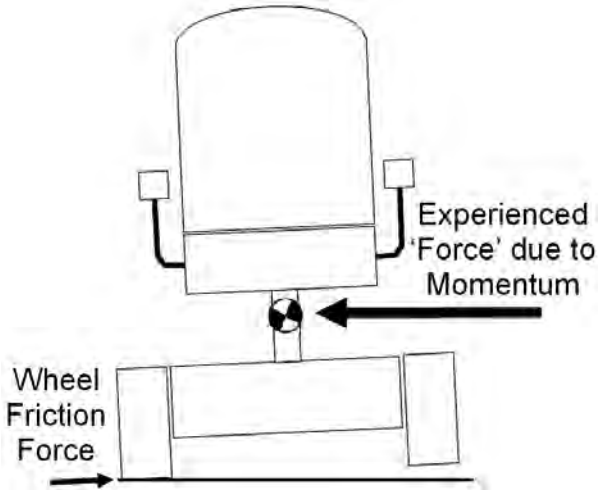

FIGURE 3.7: WHEEL LIFT DUE TO 'MOMENTUM' FORCE CAUSING OVER-STEER (WOLM ET AL., 2006)

3.4.3 Disproving the Initial Wheel Lift Theory

A roll gyroscope sensor was used specifically to measure wheel lift. The roll rate was integrated to obtain roll angle. Wheel lift heights were calculated from the known roll angles and wheelchair geometry. Figure 3.8 indicates several instances of wheel lift during a dog leg pattern with the first turn to the right and driven forward without loss of control.

The data clearly indicates that wheel lift occurred, although no lifting had been visually observed during the test run. Since there was no loss of control, it was highly unlikely that wheel lift occurred according to the wheel lift theory. To reaffirm the calculated wheel lift heights, a physical test was performed with the wheelchair using an inclinometer, car jack, and a ruler. The car jack was used to raise one side of the wheelchair, but only one of the drive wheels. The wheel was lifted until the inclinometer indicated the known angle. A measurement of the wheel clearance above the level surface with a metric ruler from the bottom of the wheel confirmed calculations. However, it was visibly evident that even a

wheel lift height of 5 mm was noticeable thus confirming that wheel lifting during test runs could not have gone unobserved.

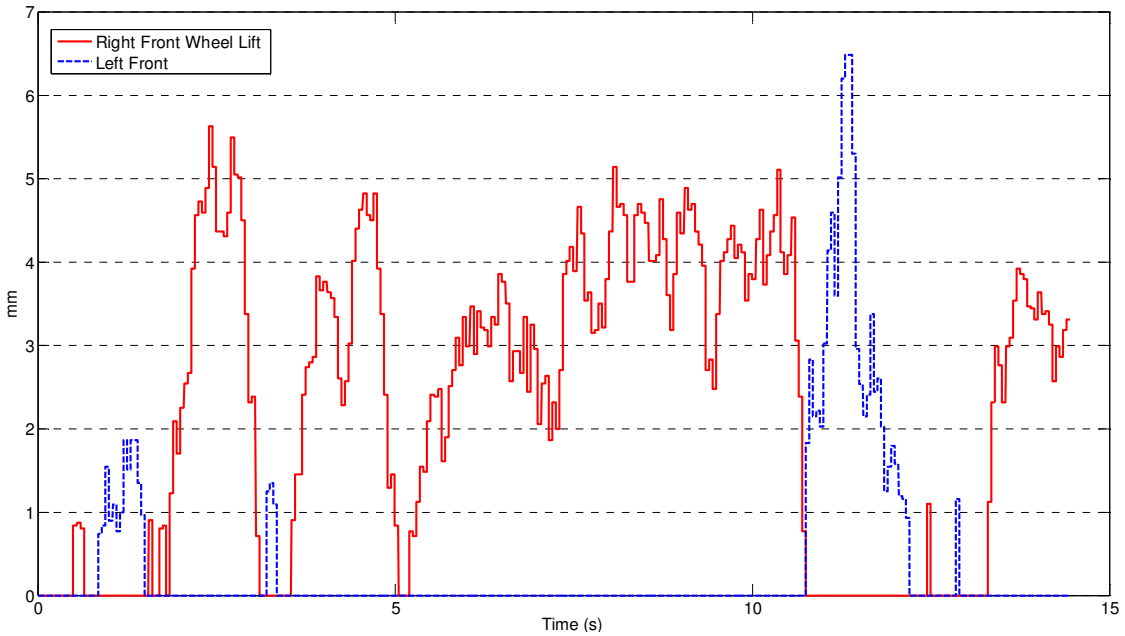

FIGURE 3.8: CALCULATED WHEEL LIFT HEIGHTS FROM INTEGRATED ROLL RATE DATA

After discussion with other users of the same model solid state gyroscope sensor, it was discovered that this specific type of gyro suffers from drift. Drift is defined as the deviation from a zero measurement when there is no rotation around the gyro sensor's axis of measurement. The drift does not affect rotation rate measurements, but becomes apparent when the rotational rates are integrated, exposing drift error. The drift error confirmed that any calculations involving integrated angles were false indicating that it was unlikely wheel lift had occurred. Even during tests where control was lost, no wheel lifting was observed by witnesses or experienced by the occupier.

Each forward direction test pattern profile was driven twice. The first test run of a profile was driven with the test-bed occupied and the second test run, unoccupied. However, each test run of a profile pattern had the same starting orientation, position, and control inputs. Figure 3.9 shows the ground path followed for both test runs of a dog leg pattern with a first right turn profile and driven under control. The figure is generated from the drive wheels' encoder data.

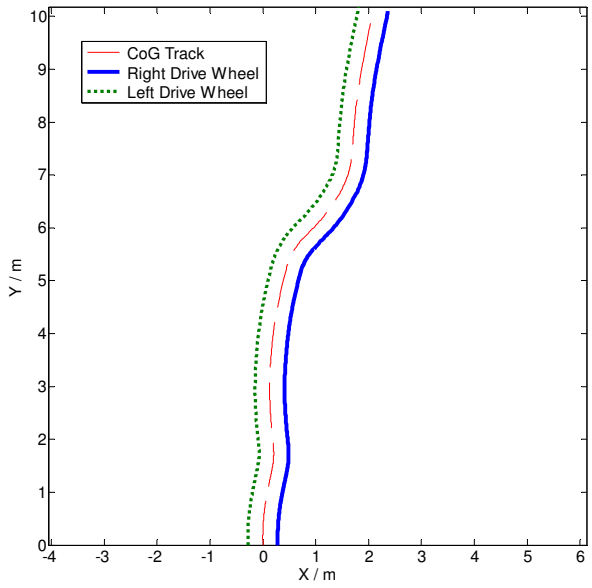

FIGURE 3.9: PLAN VIEW OF ACTUAL DRIVEN GROUND TRACK OF DOG LEG RIGHT TEST PROFILE

Final confirmation of the lack of validity of the wheel lift theory can be seen in Figure 3.10, which shows the yaw rates, or the rotational rates around the vertical axis through the CoG of the test-bed, for the dog leg right test pattern. It is clear that an occupied front wheel drive wheelchair is actually more stable for the same inputs. This difference is a result of a higher moment of inertia for the system. The CoG is approximately 5 mm farther from the line of cornering force when the test-bed is occupied. Hence, CoG position has little or no

impact on the stability when compared to the difference in inertias. It should be noted that "Pax", an acronym for passenger, has the same definition as occupier.

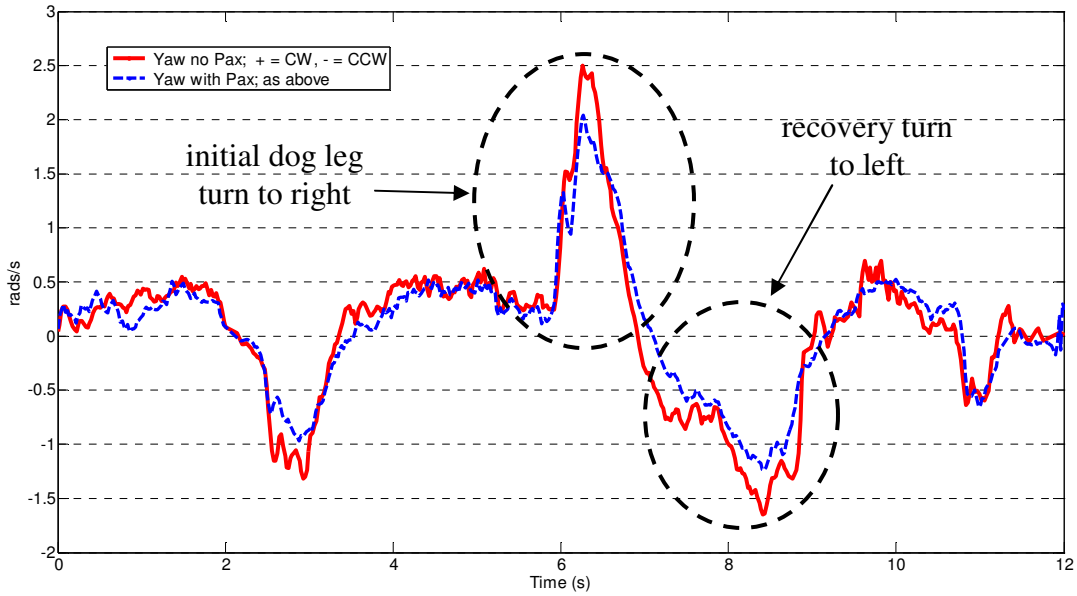

FIGURE 3.10: COMPARISON OF YAW RATE WITH AND WITHOUT OCCUPIER FOR SAME CONTROL INPUTS

3.4.4 Confirmation of Initial Proximal Theory

Referring to Figure 3.9, it can be seen that from the starting point of the test run at (0, 0) meters on the XY plane, that the test-bed wheelchair veers gently to the right followed by a sharper correction to the left. This initial turning to the right, as well as subsequent drifts from a straight line before the initial right turn of the dog leg, was the result of very small differences in wheel velocities measuring less than 0.2 m/s absolute. Although maintaining a straight line was difficult due in part to the responsiveness of the joystick, the slightest differences in wheel velocities were amplified by the over-steer instability. It should be noted that variability in motors, controllers and even wheelchair fabrication makes such small differences unavoidable.

Over-steer instability was apparent to occupants and observers of the test-bed in all forward driven test patterns in both the "under control" and "loss of control" situations. The test-bed was also driven in reverse to highlight under-steer and provide a contrast. Figure 3.11 shows a plan view of the track of a test in which the test-bed was driven in reverse with several dog leg patterns in the test run. This pattern run in reverse is considered to be a dog leg right, because the initial dog leg turns are to the right when facing the backward direction of travel. In addition, the test-bed was unoccupied in this case and was driven in a manner to cause loss of control. However, no loss of control could be achieved, as evident in Figure 3.11.

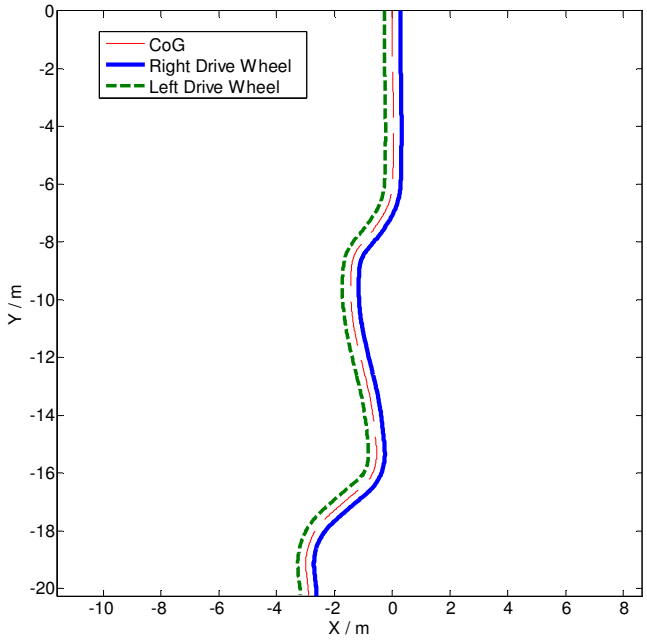

FIGURE 3.11: PLAN VIEW OF ACTUAL DRIVEN GROUND TRACK OF DOG LEG RIGHT PROFILE DRIVEN IN REVERSE

In contrast, Figure 3.12 indicates the ground track of another dog leg right test pattern driven forward, with the test-bed occupied. The goal of this particular test pattern was to cause the test-bed to lose control, which was easily accomplished. Once again, it can be

seen the initial drift to the right was corrected with a sharp turn to the left that was an overcompensation exacerbated by over-steer.

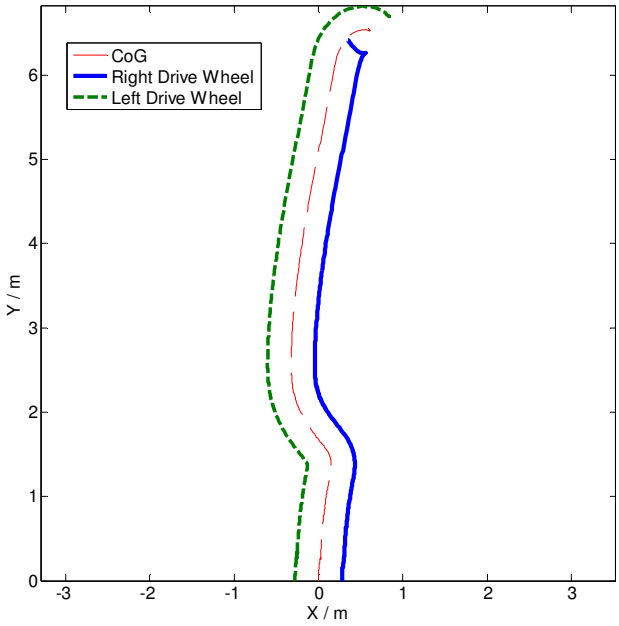

FIGURE 3.12: PLAN VIEW OF ACTUAL DRIVEN GROUND TRACK OF DOG LEG RIGHT PROFILE DRIVEN FORWARD WITH LOSS OF CONTROL

Note that the test patterns from Figure 3.9 (dog leg right, forward, occupied, no loss of control), Figure 3.11 (dog leg right, reverse, unoccupied, attempted loss of control), and Figure 3.12 (dog leg right, forward, occupied, loss of control) will be known as Patterns 1, 2, and 3 respectively for the remainder of this section.

Figure 3.13 (Pattern 1) shows the control inputs along with forward velocity and yaw rate for the test pattern driven in Figure 3.9. Referring to Figure 3.14 and Figure 3.15 (Patterns 2 and 3), it can be seen that the control inputs applied in the attempt to create a loss of control for each situation were intensive. In the case of Pattern 3, the control inputs were too severe. The 'loss of control' inputs were far greater than those used for the test profile

of Figure 3.9 (Pattern 1). Note that with Figure 3.13 to Figure 3.15, velocities are in m / s, and yaw is in radians / s. Also recall that the control inputs are duty cycle demands that decide the voltage provided to each drive motor from the typical 24 V supply.

Table 3.3 highlights the differences in the control effort to initiate dog leg turns for each test pattern. Recall that the test-bed wheelchair, as in the majority of electrically powered wheelchairs, utilises differential steering to turn. Hence, the greater the difference between the control inputs to each drive motor, the greater the turning rate induced regardless of any destabilising moment.

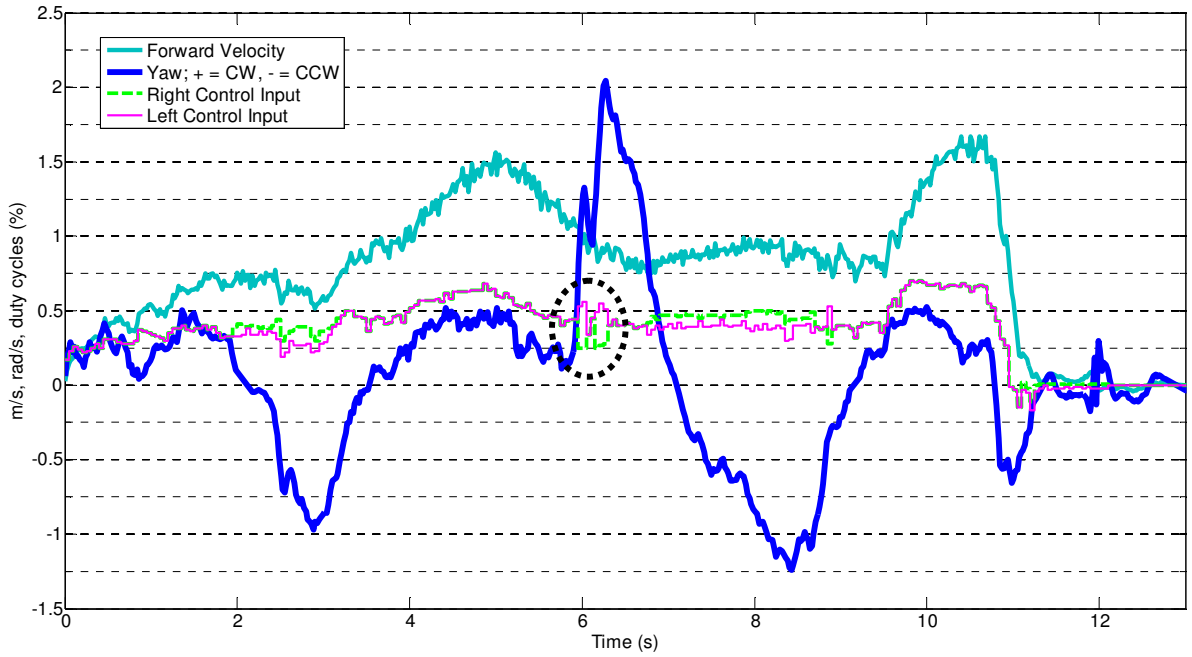

FIGURE 3.13: PATTERN 1 - YAW WITH VELOCITY FOR DOG LEG RIGHT PROFILE DRIVEN FORWARD WITHOUT LOSS OF CONTROL

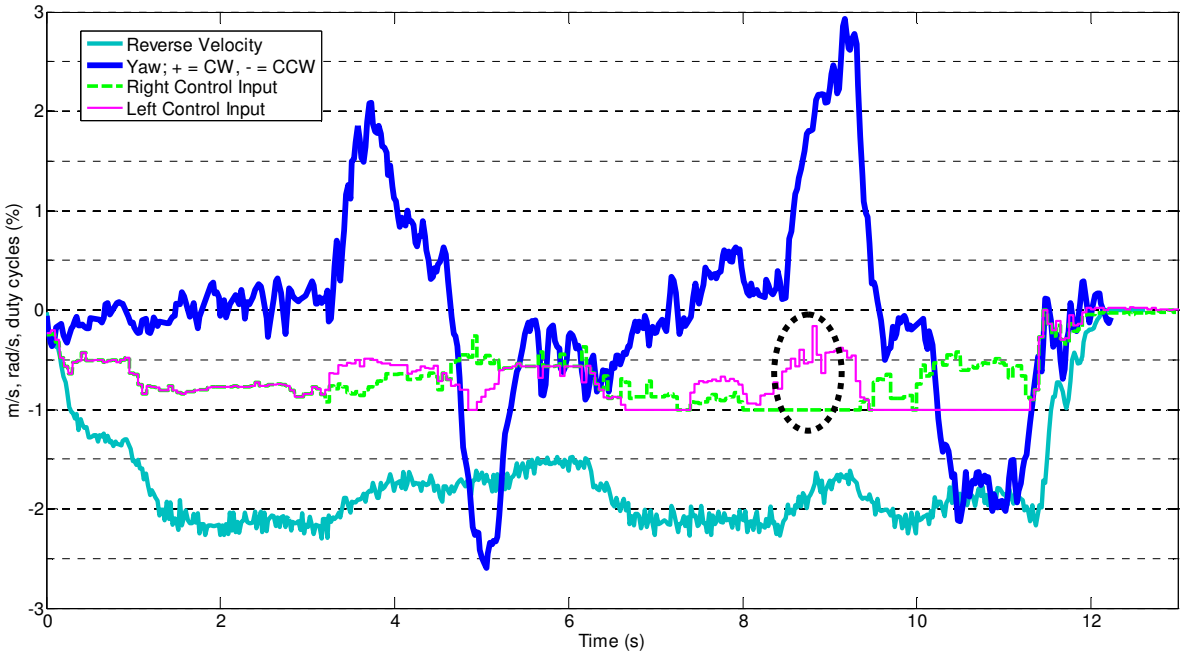

FIGURE 3.14: PATTERN 2 - YAW WITH VELOCITY FOR DOG LEG RIGHT PROFILE DRIVEN IN REVERSE

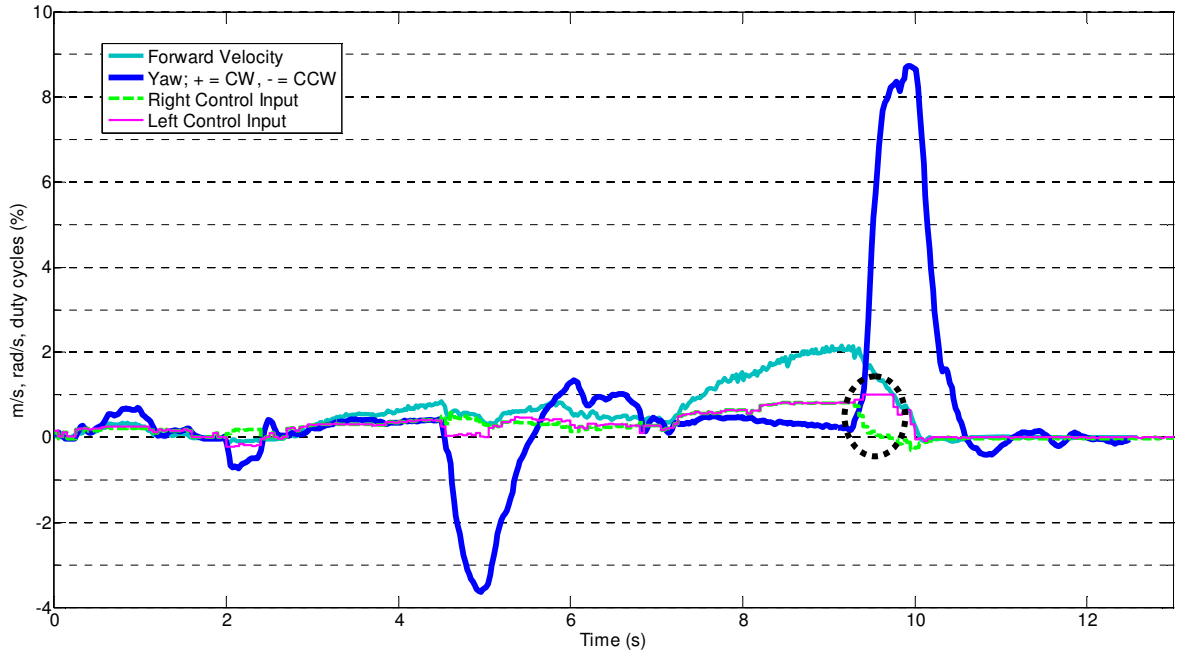

FIGURE 3.15: PATTERN 3 - YAW WITH VELOCITY FOR DOG LEG RIGHT PROFILE DRIVEN FORWARD WITH LOSS OF CONTROL

TABLE 3.3: DOG LEG RIGHT TEST PROFILES' CONTROL INPUTS FOR THE INITIAL TURN

Pattern	Condition	Result	Direction	Left Motor Control Input	Right Motor Control Input	Control Input Difference
1	Occupied	No loss of control	Forward	56	25	31
2	Unoccupied	No loss of control	Reverse	- 16	- 100	84
3	Occupied	Loss of control	Forward	100	0	100

The effect of the destabilising moment, *M* described in Section 1.3, becomes obvious between Patterns 1 and 2. For Pattern 2, since the CoG is now ahead of the line of cornering force, *M* acts as a stabilising, or even as an 'over-damping' moment, thus causing under-steer. It was observed that it was much more difficult to effect sharp turns when the test-bed was driven in reverse. For a much greater control input difference, 84 versus 31, no control is lost in reverse even though the occupied test-bed should be more stable. Furthermore, the control differences between Pattern 1 and 3 are much closer together, 84 versus 100, but complete control has been lost in Pattern 3.

When observing the velocities involved in each of the 3 patterns, it can be seen that a higher velocity overall was maintained in Pattern 2 than either Patterns 1 or 3. The velocity has an effect on *M*, as velocity affects momentum, which in turn affects the cornering force. In Pattern 1 it was necessary to drive more slowly to maintain control. In pattern 3, the entry velocity before the initial dog leg turn was over 2 m/s. Forward velocity naturally fell

away during loss of control. However, in Pattern 2, with entry velocities even greater than that of Pattern 3, this behaviour had no effect on controllability.

3.5 SUMMARY

Data from live tests of the wireless prototype test-bed were successfully captured and processed. The live tests followed set patterns known as doglegs and ovals. Dogleg patterns in particular are known to emphasise the over-steer properties found in FWD wheelchairs. The methods used to process the data are described as are the specific technical difficulties or issues uncovered during data collection.

Data analysis confirmed one theory explaining over-steer by Collins and Kauzlarich [3], while refuting another postulated during this research. The postulated initial theory espoused that over-steer instability would be increased with an occupied wheelchair versus and empty one. This initial theory explained that an occupied wheelchair's CoG is higher than otherwise, thus increasing the chance of wheel lift and the resulting over-steer. However, the results indicated the opposite as an occupant increases the stability of FWD wheelchairs with the increased mass moment of inertia.

Comparisons were also made between the results from tests in which the test-bed was driven forward and in reverse. A FWD wheelchair acts as a RWD wheelchair when driven in reverse and in that case suffers from under-steer thus confirming the theory found in [3].

4

WHEELCHAIR SYSTEM MODELLING

4.1 INTRODUCTION

Generally, system models have been created to overcome costs, time, and hazards of analysing actual live systems. The FWD wheelchair model inherited from the 2006 project team featured kinematic equations that were transferred into a dynamic model using Mathworks' Simulink graphic software. The prime aim of the validated model is to be a single input parametric model that can be adjusted for each unique FWD wheelchair system.

In this case, even though there are actually two inputs to the model, the left and right drive motor control inputs, it is considered a single input in that no other forces on the wheelchair are known beforehand except those drive forces. Additionally, those drive forces are a function of the known inputs and are thus related. Finally, the possibility to apply the wheelchair model to other drive type wheelchairs, RWD and MWD, is considered.

The initial model has been enhanced and changed in this research. For instance, the method of control inputs has been changed to reflect the characteristics of the actual motor

controller. In addition, the model sub-system that sought to measure wheel lift has been eliminated as a result of data analysis. However, the final model has been altered substantially from the original base dynamic equations. All enhancements and changes are discussed in detail in Chapter 6, Model Analysis. The following pages describe the systems of the original model, a tidier version of which can be seen in Figure 4.1.

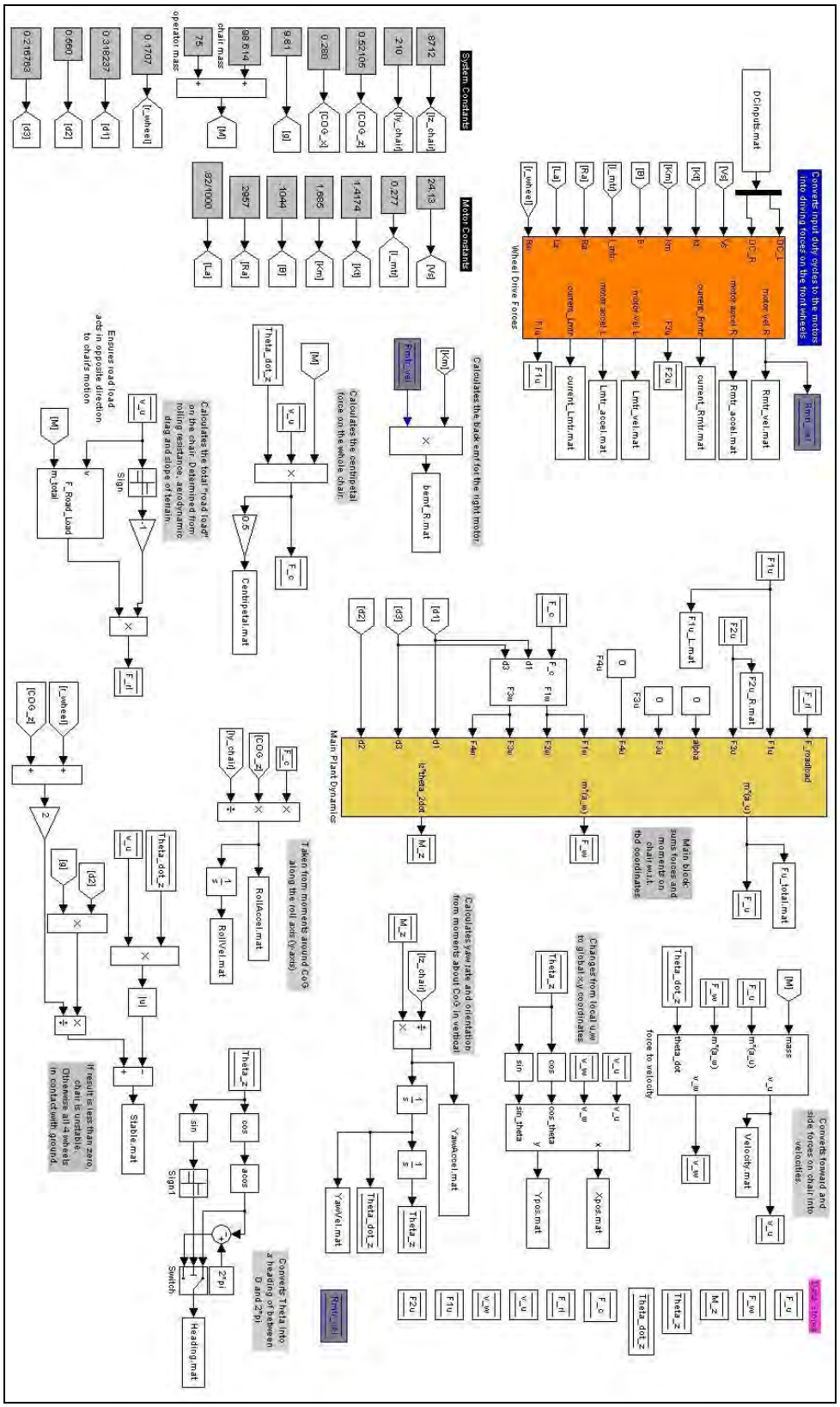

FIGURE 4.1: FWD SIMULINK MODEL INHERITED FROM 2006 RESEARCH GROUP

4.2 MAIN SYSTEM MODELLING

Recall from Section 1.4.2 that all the equations used to describe the FWD model of the test-bed were obtained from Ding et al. [6]. The equations describe both kinematic and dynamic motion. In particular, the dynamic equations, which are based on Newton's Second Law of Motion, are developed from the free body diagram shown in Figure 4.2.

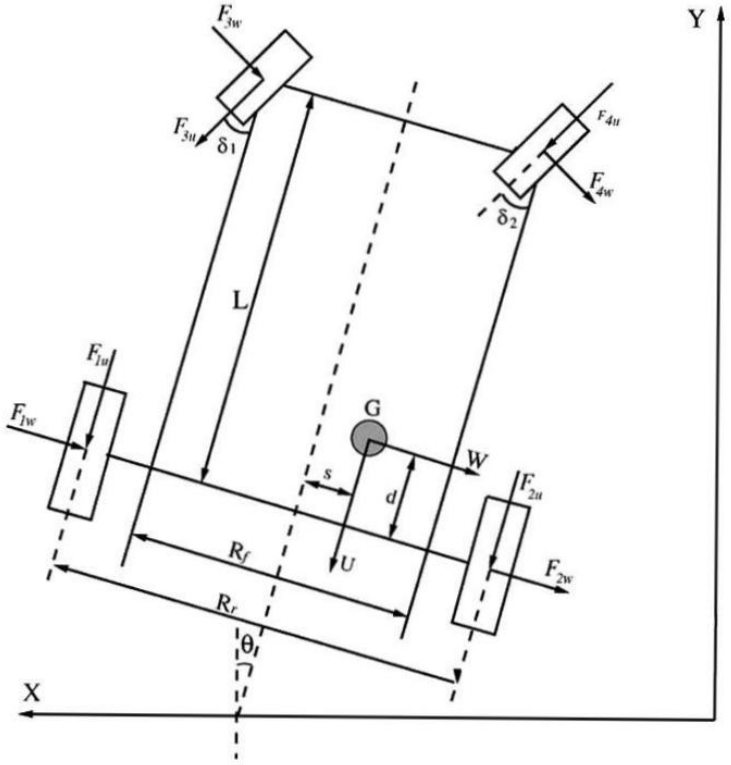

FIGURE 4.2: FREE BODY DIAGRAM OF RWD WHEELCHAIR (DING ET AL., 2004)

Referring to Figure 4.2, the nomenclature is defined:

U = local co-ordinate axis in direction of travel

W = local co-ordinate axis in direction perpendicular to U

X = global co-ordinate axis in lateral direction

Y = global co-ordinate axis in longitudinal direction, perpendicular to X

G = CoG

L = wheelbase, the distance between the ground contact points of a front drive wheel and a rear caster on the same side of the wheelchair

d = distance to the CoG from the line of cornering force, or the connecting line between the ground contact points of the two drive wheels, in the U axis direction

s = lateral distance of the CoG to the physical centre line of the wheelchair in the W axis direction

R_f = distance between ground contact points of the two casters

R_r = distance between ground contact points of the two drive wheels

$\delta_{1,2}$ = angle of each caster respectively to the U axis

θ = angle between local axes and global axes

$F_{1u,2u}$ = reaction forces at drive wheels in the U axis direction

$F_{1w,2w}$ = reaction forces at drive wheels in the W axis direction

$F_{3u,4u}$ = reaction forces at casters in the U axis direction

$F_{3w,4w}$ = reaction forces at casters in the W axis direction

When a RWD wheelchair is driven in reverse its dynamic characteristics mirror those of a FWD wheelchair driven forward. In both situations, the caster wheels are trailing the direction of motion. Due to the symmetry of characteristics of RWD and FWD wheelchairs when driven respectively forward and reverse, it was deemed that the equations provided by Ding et al. would adequately describe a FWD model and the effects of over-steer instability.

4.2.1 Kinematic Equations

According to Ding et. al [6], the kinematic model makes the assumption that no slip occurs. This assumption is in line with this research, as any destabilising moment can only exist until slip is experienced. As a result of this assumption, simple rigid body kinematics can be used to describe a wheelchair's motion. Therefore, given the drive wheel rotational velocities, ω_L and ω_R, the rotational velocity or yaw rate, $\dot{\theta}$, around the vertical axis through the wheelchair CoG, the velocity in the direction of wheelchair travel, v_u, and the velocity perpendicular to the direction of travel, v_w, can be calculated as follows:

$$\dot{\theta} = \frac{R}{R_r}(\omega_L - \omega_R) \tag{4.1}$$

$$v_u = (\omega_R + \omega_L)\frac{R}{2} - \dot{\theta}s \tag{4.2}$$

$$v_w = \dot{\theta}d \tag{4.3}$$

where:

R = radius of drive wheel

The velocity of the wheelchair for all tests was calculated with Equation 4.1. However, s = Ø was assumed as the CoG in the lateral direction W was only ≈ 2 mm from the physical centre line of the test-bed. The rotational velocities of the drive wheels, ω_L and ω_R, were calculated from the measurements provided by the drive wheel encoder sensors that delivered a count value.

The count value was first converted to angular rotation with Equation (3.3) from Section 3.3, *Step 5*. Then, the angular rotations were converted to rotational velocities as all

angular changes were time stamped. Hence, the time difference between changes in angular rotation could be calculated, thus providing $\theta/\Delta t$.

4.2.2 Dynamic Equations

Referring to Figure 4.2, given expressions describing the reactive forces can be obtained with $s = \emptyset$ assumed. In keeping with Newton's Law, the sum of the forces in the U, W and around the Normal axis directions are as follows:

$$ma_u = F_{1u} + F_{2u} + F_{3u}\cos\delta_1 + F_{4u}\cos\delta_2 + F_{3w}\sin\delta_1 + F_{4w}\sin\delta_2 \tag{4.4}$$

$$ma_w = F_{1w} + F_{2w} + F_{3w}\cos\delta_1 + F_{4w}\cos\delta_2 - F_{3u}\sin\delta_1 - F_{4u}\sin\delta_2 \tag{4.5}$$

$$\begin{aligned} I_z\ddot{\theta} = {} & (F_{1u} - F_{2u})\frac{R_r}{2} - (F_{1w} + F_{2w})d + (F_{4u}\cos\delta_2 + F_{4w}\sin\delta_2)\frac{R_r}{2} \\ & - (F_{3u}\cos\delta_1 + F_{3w}\sin\delta_1)\frac{R_r}{2} \\ & + (F_{4w}\cos\delta_2 - F_{4u}\sin\delta_2 + F_{3w}\cos\delta_1 - F_{3u}\sin\delta_1)(L - d) \end{aligned} \tag{4.6}$$

Equations (4.4) to (4.6) are located within the "Main Plant Dynamics", which is a single block within the model of Figure 4.1.

4.3 MODEL SUB-SYSTEMS

There is no description for those systems that calculate roll around the longitudinal axis and wheel lift, as these systems have been proven to be inconsequential in Chapter 3. Also, there is no description for the system that calculates wheelchair heading as this system is

redundant. A continuous and X and Y position function serves the purpose of indicating heading indirectly. The following sub-systems that are described have been maintained in one form or other in later enhancements and can also be found in Figure 4.1.

4.3.1 Wheelchair Velocity

Once again referring to Figure 4.2 and relating the local to the global co-ordinates provides:

$$v_x = v_u \sin\theta - v_w \cos\theta \tag{4.7}$$

$$v_y = -v_u \cos\theta - v_w \sin\theta \tag{4.8}$$

Expressions for a_u and a_w are obtained by differentiating (4.7) and (4.8)

$$\dot{v}_x = \left(\dot{v}_u + v_w\dot{\theta}\right)\sin\theta - \left(\dot{v}_w - v_u\dot{\theta}\right)\cos\theta \tag{4.9}$$

$$\dot{v}_w = -\left(\dot{v}_u + v_w\dot{\theta}\right)\cos\theta - \left(\dot{v}_w - v_u\dot{\theta}\right)\sin\theta \tag{4.10}$$

where it can be seen by comparing (4.7) and (4.8) with (4.9) and (4.10) respectively that

$$a_u = \dot{v}_u + v_w\dot{\theta} \tag{4.11}$$

$$a_w = \dot{v}_w - v_u\dot{\theta} \tag{4.12}$$

Solutions to Equations (4.4) and (4.5) contained in the "Main Plant Dynamics" block were passed to the sub-system entitled "force to velocity" and containing (4.11) and (4.12). Thus, the coupled equations were used to solve for v_u and v_w, the wheelchair's forward and lateral velocity respectively.

4.3.2 Road Load

The road load sub-system was meant to provide opposition to the drive forces due to rolling friction, wind resistance and the possibility of slopping inclines with the following equation

$$F_{RL} = \mu_r N + \frac{1}{2}\rho v^2 C_d A + N \sin\theta \tag{4.13}$$

where:

μ_r = rolling resistance coefficient

N = normal force of the wheelchair system

ρ = air density

v = forward velocity of the wheelchair

C_d = distance between ground contact points of drive wheels

A = frontal area of wheelchair system

The results of solving Equation (4.13) contained in the block with the output labelled "F_Road_Load" were added to Equation (4.4) in the "Main Plant Dynamics" block. Additional Simulink symbols, such as "Sign" ensured that the road load always acted in opposition to the direction of travel.

4.3.3 Lateral Cornering Forces

In keeping with the aim of creating a single input model another means was needed to calculate the lateral forces of the FWD wheelchair model. A standard equation to calculate the centripetal force on a vehicle through its CoG in a turn is used, which is defined:

$$F_c = \frac{mv_u^2}{r} \tag{4.14}$$

where r is the radius of curvature of the turn. However, r was not directly available in the original model from the dynamic system equations. Thus, Equation (4.14) was transformed using available information to be defined:

$$F_c = mv_u\dot{\theta} \tag{4.15}$$

With the CoG location and the dimensions of the wheelchair model fixed, the values of the lateral forces were derived through statics and taking moments around the appropriate axes. Thus, the lateral wheel loads could be calculated in the model from the existing dynamic system equations. As can be seen in Figure 4.3, the centripetal force, F_c or F_c, calculated separately using Equation (4.15), is passed to the indicated block. In this case, d3 represents d, d1 represents L - d, and d2 represents R_r, all found in Equation (4.6).

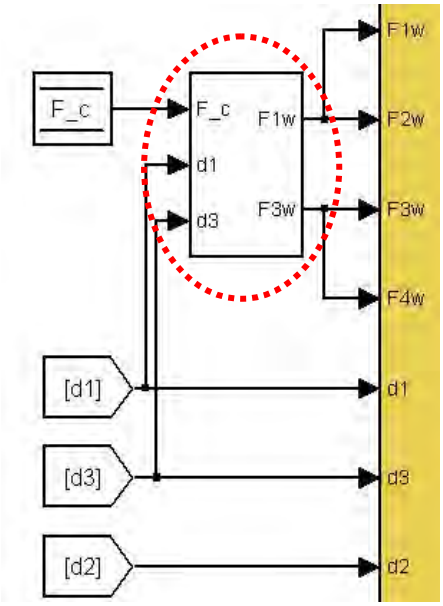

FIGURE 4.3: BLOCK IN 2006 SIMULINK MODEL WHERE FINAL DERIVATION OF LATERAL FORCES ON ALL WHEELS CALCULATED

4.3.4 Yaw Velocity

Yaw velocity is obtained by dividing the result of Equation (4.6) by the inertia of the test-bed wheelchair about its vertical axis "Iz_chair". The resulting $\ddot{\theta}_z$ value is integrated twice. First to obtain $\dot{\theta}_z$, the yaw velocity of the wheelchair, and then, θ_z, the directional orientation of the wheelchair.

4.3.5 Position Plotting

This sub-system is used obtain the wheelchair position for plotting in the plan view using X and Y co-ordinates. The results for v_u, v_w, and θ_z are applied to Equations (4.7) and (4.8). The results from these equations are integrated to obtain the X and Y co-ordinates of the wheelchair for each time-step of the simulation.

4.3.6 PM DC Motor Model System

The permanent magnet direct current (PM DC) motors found on the test-bed, which are typical to EPWs, are the main means of providing input forces. Hence, an accurate motor model is required as the resulting wheelchair dynamics are a consequence of the inputs from these modelled motors. The standard expressions provided by Rizzoni [22] describing a DC motor, are defined by two coupled ordinary differential equations.

Using Kirchoff's voltage law and summing the voltages throughout a simple circuit of a DC motor drive system, seen in Figure 4.4, results in a drive voltage defined:

$$V = R_a i + L_a \frac{di}{dt} + V_{emf} \tag{4.16}$$

where R_a, L_a and V_{emf} are the armature (or rotor) resistance, armature inductance and the back electro-motive force. The equation of motion for a DC motor drive, ignoring any load, gives torque as a second order ODE in rotation defined:

$$T = J\ddot{\theta} + B\dot{\theta} \tag{4.17}$$

where J and B are the motor inertia (or equivalent system inertia in a geared motor system reflected on the motor shaft) and the viscous friction coefficient, also know as viscous damping coefficient (or equivalent viscous friction for a geared motor system).

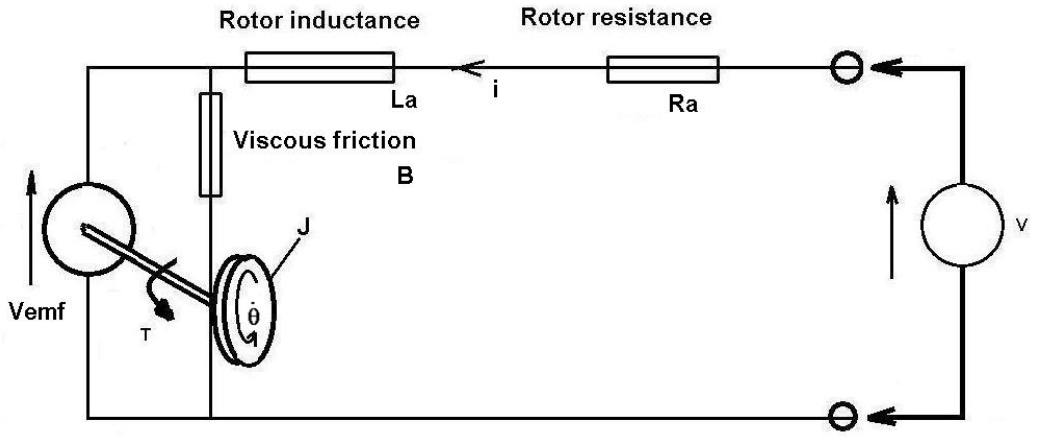

FIGURE 4.4: SIMPLE DC DRIVE MOTOR CIRCUIT

The electrical and mechanical components are coupled in two ways. First, the back emf, V_{emf}, in the motor is linearly related to the motor rotational velocity:

$$V_{emf} = k_e\dot{\theta} \tag{4.18}$$

where k_e is variously known as the speed, electrical, motor or back emf constant. Second, an approximate relation generally describes motor torque as a linear function of current in the motor:

$$T = k_t i \tag{4.19}$$

where k_t is the motor torque constant. The combined electrical and dynamic relationships result in a system of equations that govern a DC motor system's response. Equations (4.18) and (4.19) are substituted into Equations (4.16) and (4.17) respectively to result in the final system equations of a PM DC motor model. In particular, a set of coupled 1st and 2nd order ODEs that can yield two results, i and $\dot{\theta}$, and are defined:

$$V = R_a i + L_a \frac{di}{dt} + k_e \dot{\theta} \tag{4.20}$$

$$k_t i = J\ddot{\theta} + B\dot{\theta} \tag{4.21}$$

Representing Equations (4.20) and (4.21) in a model of state space form provides

$$\begin{bmatrix} \ddot{\theta} \\ \frac{di}{dt} \end{bmatrix} = \begin{bmatrix} \frac{-B}{J} & \frac{k_t}{J} \\ \frac{-k_e}{L_a} & \frac{-R_a}{L_a} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L_a} \end{bmatrix} V \tag{4.22}$$

$$\dot{\theta} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} \tag{4.23}$$

The state space system of (4.22) is instituted using a standard Simulink state-space block within the "Wheel Drive Forces" block found in Figure 4.1 to provide 2 PM DC drive motors. The state-space block provided the output of Equation (4.23). The .mat file, "DCinputs.mat" contains the duty cycle requests for each motor usually recorded from live tests. The duty cycle percentage is multiplied by the supply voltage V_s of the wheelchair test-bed measured during testing, the product of which is the input voltage to each drive

motor represented by the state space system for each control demand. The $\dot{\theta}$ output is differentiated to obtain $\ddot{\theta}$ and is used by the following formula to find *F*, defined:

$$F = \frac{\tau}{R} = \frac{\ddot{\theta} J}{R} \tag{4.24}$$

Equation (4.24) is used to calculate the output forces of the drive wheels. The parameters *J* and *R* are the motor system inertia and radius of the drive wheel.

4.4 SUMMARY

A mathematical dynamic system model of a FWD wheelchair, inherited from previous research, is described in this chapter. The dynamic model contains dynamic equations of motion derived from first principles that describe the wheelchair motion, including the effects of over-steer and the mass moment of inertia. It also includes equations that describe the dynamics of the PM DC drive motors that govern the force inputs to the system. The dynamic model has been altered slightly due to the results of the data analysis in Chapter 3, which eliminated a certain sub-system in the model defining wheelchair roll about the model's longitudinal axis. Further, a basic inspection resulted in the removal of a redundant sub-system that calculated heading, while retaining the system that plotted the wheelchair model's X and Y position in plan view.

5

PARAMETER VALUES

5.1 INTRODUCTION

The model parameters required can be placed into two categories: 1) System Parameters, and 2) PM DC Motor Parameters. The System Parameters are used in the dynamic equations first mentioned in Chapter 4, which are presented here for review.

$$ma_u = F_{1u} + F_{2u} + F_{3u}\cos\delta_1 + F_{4u}\cos\delta_2 + F_{3w}\sin\delta_1 + F_{4w}\sin\delta_2 \tag{5.1}$$

$$ma_w = F_{1w} + F_{2w} + F_{3w}\cos\delta_1 + F_{4w}\cos\delta_2 - F_{3u}\sin\delta_1 - F_{4u}\sin\delta_2 \tag{5.2}$$

$$\begin{aligned} I_z\ddot{\theta} = {} & (F_{1u} - F_{2u})\frac{R_r}{2} - (F_{1w} + F_{2w})d + (F_{4u}\cos\delta_2 + F_{4w}\sin\delta_2)\frac{R_r}{2} \\ & - (F_{3u}\cos\delta_1 + F_{3w}\sin\delta_1)\frac{R_r}{2} \\ & + (F_{4w}\cos\delta_2 - F_{4u}\sin\delta_2 + F_{3w}\cos\delta_1 - F_{3u}\sin\delta_1)(L-d) \end{aligned} \tag{5.3}$$

The required parameters for the PM DC motors are used in the Equation (5.4), also previously detailed in Chapter 4.

$$\begin{bmatrix} \ddot{\theta} \\ \dfrac{di}{dt} \end{bmatrix} = \begin{bmatrix} \dfrac{-B}{J} & \dfrac{k_t}{J} \\ \dfrac{-k_e}{L_a} & \dfrac{-R_a}{L_a} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{1}{L_a} \end{bmatrix} V \tag{5.4}$$

The following pages describe those values that have been used in the enhanced dynamic system model and how they were measured and calculated.

5.2 SYSTEM PARAMETERS

All the system parameter measurements are described in detail by Wolm et al. [28]. In this research, the CoG parameters have been reconfirmed. In particular, the validity of the value of the moment of inertia through the vertical axis value has been questioned and thus re-examined. Table 5.1 provides the values of the parameters that have been retained in the enhanced model, either from the prior work or adjusted during this research.

TABLE 5.1: SYSTEM PARAMETER VALUES

Parameter		Value
m	Wheelchair Mass	98.614 kg
I	Wheelchair inertia around vertical axis	5.5 kg•m^2 (estimated)
R_r	Distance between drive wheels	0.56 m
d	Distance of CoG from line of cornering force	0.2168 m
R	Drive wheel radius	0.1717 m

5.2.1 Wheelchair Mass, m

A load cell suspended from a crane was used to find the mass of the wheelchair by suspending the wheelchair levelly from the load cell and calculating the mass from the load cell measurement. If the wheelchair model is to have a passenger then the mass, *m*, in this case reflects the entire mass of the system, both the wheelchair test-bed and passenger. Figure 5.1 shows the mass measuring equipment format. The mass of the wheelchair was adjusted with the addition of the new equipment described in Chapter 2 under the Current Form sections.

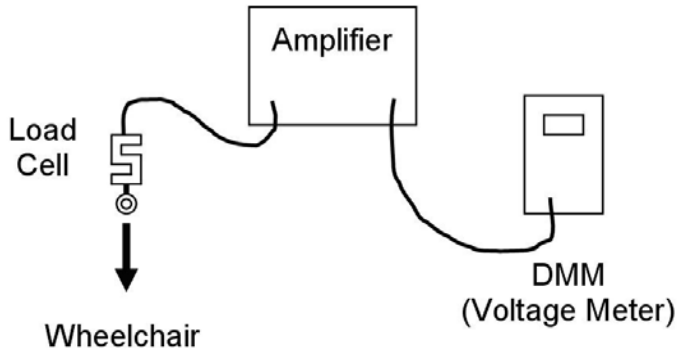

FIGURE 5.1: LOAD CELL MASS MEASURING SYSTEM (WOLM ET AL., 2006)

5.2.2 CoG Parameters, d

The same equipment and crane configuration that was used to measure the wheelchair mass was also utilised to determine the CoG location and hence *d*, defined in Table 5.1. A leverage principle was used by lifting each side of the wheelchair and measuring weight distribution as shown in Figure 5.2.

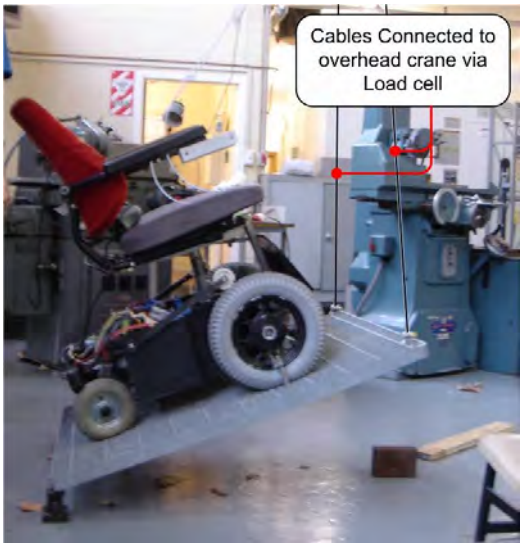

FIGURE 5.2: FINDING COG THROUGH LEVERAGE (WOLM ET AL., 2006)

The accuracy of the previously calculated CoG position was reconfirmed in this research by placing each wheel of the wheelchair test-bed on 4 separate scales simultaneously and reapplying the leverage principle.

5.2.3 Mass Moment of Inertia, I

A quadrifilar pendulum, as seen in Figure 5.3, was used to measure the mass moment of inertia, *I*. This type of pendulum has been utilised to measure the inertia of non-standard shapes and offered the best hope of calculating the most accurate moment of inertia. A formula presented by Steidel [24] for a bifilar pendulum was adjusted for measurements using a quadrifilar pendulum. However, it was felt that this experimental result was suspect in its accuracy although there has been no means available until recently for a viable comparison of results.

FIGURE 5.3: USING QUADRIFILAR PENDULUM TO FIND WHEELCHAIR TEST-BED MOMENT OF INERTIA (WOLM ET AL., 2006)

Reference has been made to the inertias' of several EPWs [6] of similar dimensions and mass to the wheelchair test-bed but the inertias were estimated through an experimental method which has not been detailed and each EPW inertia test contained an additional 75 kg human dummy. Furthermore, the units used to describe the inertias are *kg•m*, an unfamiliar unit for which no clarification has been obtained.

However, Wang et al. [27] has also made reference to [6] and appear to have clarified the unusual units. They found that the moments of inertia, presumably for empty EPWs, are similar to a result obtained by using another quadrifilar pendulum method outlined in [27]. Accordingly, the clarification of inertia values found in [6] of 6 and 8 kg•m^2, and the result in [27] of 5.228 kg•m^2 indicate that the experimental result for inertia of the wheelchair test-bed stated in [28] of 0.871 kg•m^2 is indeed likely to be inaccurate.

Correspondence with Professor Raul G. Longoria (Department of Mechanical Engineering, University of Texas at Austin) also confirmed that the inertia result of 0.871 kg•m^2 was most likely highly inaccurate. The measurements and subsequent calculations for inertia using filar pendulums require the motion of the pendulum to be a precise, periodic, harmonic motion. Professor Longoria confirmed that when a non-instrumented observation method is used the results are highly inaccurate because small a-periodic motion with several higher harmonics that occur regularly in filar pendulum experiments cannot be observed and also cannot be discounted. Wolm et al. [28] used non-instrumented visual observation to obtain measurements and further opportunity to repeat the quadrifilar experiment with instrumentation has not been available. However, it is most likely that the test-bed's moment of inertia around its vertical axis can be estimated with a near proximity using the values of [6] and [27]. Therefore, an estimated a value of 5.5 kg•m^2 has been chosen in this research as reported in Table 5.1.

5.2.4 General dimensions

Further measurements were made, such as drive wheel diameter mentioned in Table 2.1, using metric rulers and calipers to obtain the drive wheel radius R and the distance between the drive wheels R_r, both of which can be seen in Table 5.1.

5.3 PM DC MOTOR PARAMETERS

All the PM DC motor parameter measurements were presented in [28]. They were repeated in this research, as detailed by Wolm et al. [29] to reconfirm the values. To find the

required PM DC motor parameters one of the drive motors was removed from the test-bed. The drive motor was part of an entire drive system that also contained a gear box and the wheel rim and tyre. Hence, certain parameters are lumped, such as the motor moment of inertia, which now contains the inertias of the gear box and wheel rim as well as the motor armature. Table 5.2 provides the values of the identified parameters.

TABLE 5.2: PM DC MOTOR SYSTEM PARAMETER VALUES

Parameter		Value
k_t	Torque constant	1.4882 Nm/A
k_e	Speed constant	1.685 rads/s/V
R_a	Armature resistance	0.2957 Ω
B	Viscous friction coefficient	0.1044 Nm•s/rad
L_a	Armature inductance	.082 mH
J	Drive system inertia	0.270 kg•m^2

5.3.1 Torque constant, k_t

Rearranging the Equation (4.19) provides the torque constant in terms of torque versus current. Hence, obtaining k_t requires concurrent torque and current measurements. Figure 5.4 shows the test arrangement for this torque measurement. It consists of a geared PM DC motor drive system, two Newton force meters, a digital multi-meter (DMM) to measure voltage at the motor terminals, a DMM clamp meter to measure current, and a rope to provide a load on the motor.

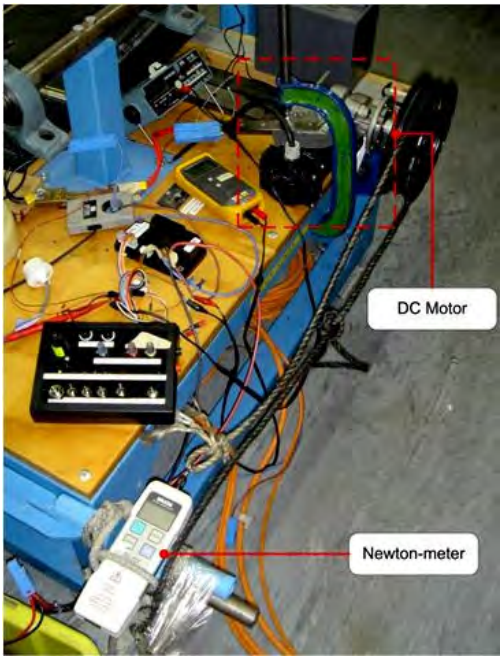

FIGURE 5.4: TEST CONFIGURATION TO MEASURE MOTOR TORQUE CONSTANT (WOLM ET AL., 2006)

A steady 24 Volts was supplied to the motor, and the motor was allowed to reach a constant speed. The load was steadily increased until motor stall. Motor current, the forces on both ends of the rope, and motor terminal voltage were measured. Motor terminal voltage was measured to ensure that voltage was steady at the motor terminals. The motor torque was calculated by multiplying the difference between the two forces on each end of the rope with the radial distance between wheel hub and where the rope rested on the wheel rim such that $T = FR$. Figure 5.5 shows the results with the slope of the line providing the torque constant the value of which is given in Table 5.2.

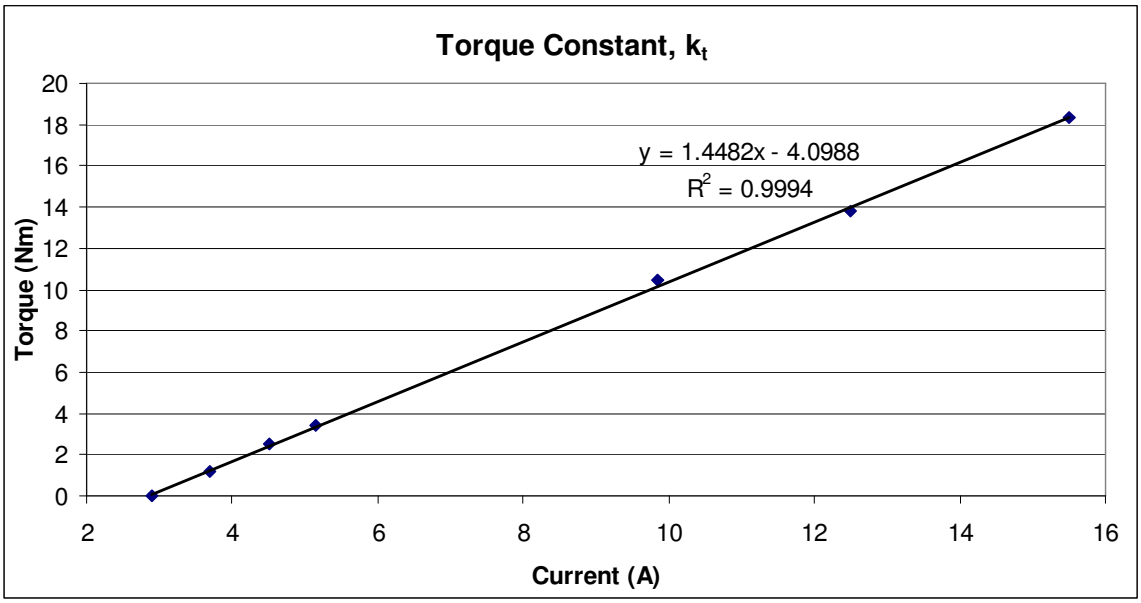

FIGURE 5.5: RESULTS OF TORQUE VERSUS CURRENT TEST, WHERE K_T = SLOPE

5.3.2 Speed constant, k_e

If only looking at steady state values for current, rearranging Equation (4.20), to give a steady state slope-intercept form, yields:

$$\frac{V}{i} = k_e\left(\frac{\dot{\theta}}{i}\right) + R_a \tag{5.5}$$

where $\dot{\theta}$ is, in this case, steady state rotational velocity of the motor rotor shaft. From Equation (5.5) only steady state values of input voltage, current and motor shaft velocity are needed. The test set-up included the same geared PM DC motor drive system, a steady 24 V supply, DMM to measure terminal voltage, DMM clamp meter to measure current, a means to control motor velocity or the voltage at the motor terminals, which in this case was a variable resistance, and a digital rpm gauge that measured the rpm of the wheel which was assumed to equal the motor shaft velocity. Figure 5.6 provides the results of

measurements made at multiple terminal voltage inputs under "no load" conditions with k_e equalling the slope of the resulting line and the value of which is given in Table 5.2.

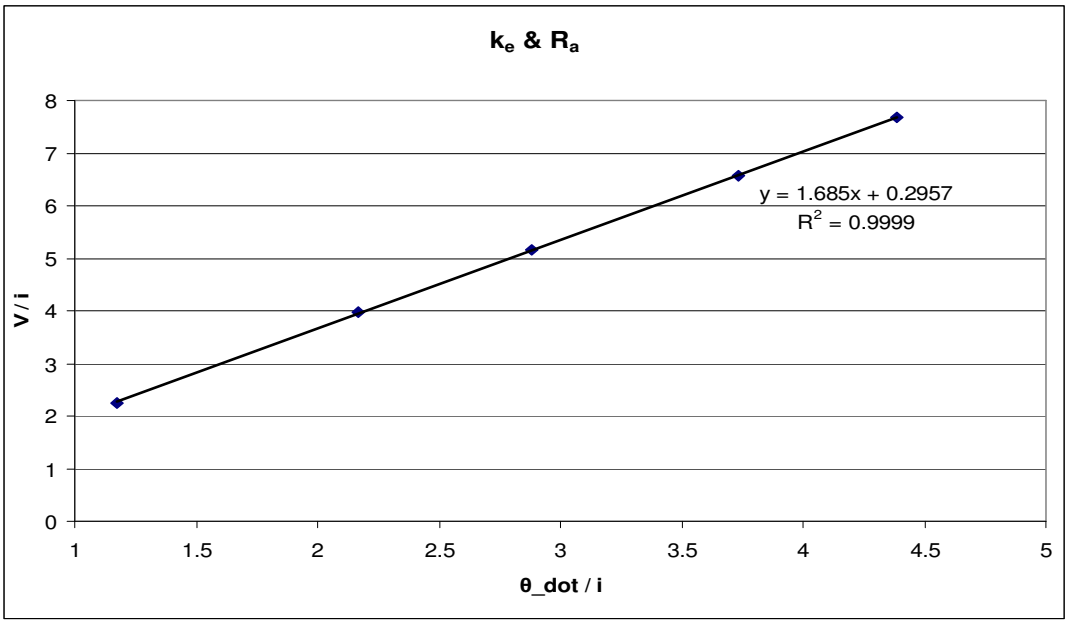

FIGURE 5.6: STEADY STATE RESULTS OF EQUATION (5.5) WHERE K_E = SLOPE

It should be noted that in an ideal DC motor, the speed constant is equal to the torque constant. However, a significant difference arises particularly in a geared motor system due to new additional external factors introduced by an added gear box. This introduction effects the torque constant, k_t, however, the speed constant, k_e, is still directly dependent upon the PM DC motor characteristics such as the strength of the permanent magnets. Hence, the difference in values as reported in Table 5.2

5.3.3 Armature Resistance, R_a

By utilising Equation (5.5) and the results for the speed constant, the armature resistance can be taken from the equation of the line. Specifically, this value is the y axis intercept value as seen in Figure 5.6 and given in Table 5.2.

5.3.4 Viscose Friction Coefficient, B

Under steady state rotational velocity Equation (4.21), becomes:

$$k_t i = B\dot{\theta} \tag{5.6}$$

Rearranging Equation (5.6) provides:

$$B = k_t i / \dot{\theta} \ \underline{\text{OR}} \ B = k_t i / \omega \tag{5.7}$$

A range of steady state current and corresponding rotational velocity values were measured with the same set-up used to calculate the speed constant. This range of values was used to calculate B, such that i/ω became $\Delta i/\Delta\omega$, with k_t already known. The friction coefficient also takes into account the gearing and eddy current losses in the motor iron, which increase with speed. Hence, B tends to be a 'lumped' parameter incorporating more than one specific dynamic behaviour, the value of which is given in Table 5.2.

5.3.5 Armature Inductance, L_a

To measure the armature inductance a digital meter capable of measuring and recording instantaneous or real-time changes was needed. The instrument used was a Fluke 105B Scopemeter. The test set-up of the PM DC motor drive system also required a steady DC voltage supply. A fixed voltage was supplied to the at-rest motor system, with the

rotor/wheel locked. The rate of current rise with time was captured by the meter and inductance was calculated using Equation (5.8) following this paragraph. Back emf, V_{emf}, is zero since there is no rotor rotation. Initial current is zero and subsequently negligible when compared with rate of change of current within the first instance of motor start-up. As a result, Equation (4.20) becomes, when rearranged:

$$L_a = V / \left(\frac{di}{dt} \right) \tag{5.8}$$

Within the first 3 ms the current went from 0 to 44 A with a 12 V step input providing the final value for L_a as reported in Table 5.2.

5.3.6 Drive system inertia, J

Once again the real-time measuring meter, the Fluke 105B Scopemeter was employed and the same test set-up was used as for measuring armature inductance, but with a means to control voltage input and thus motor current levels. In this case, the rate of voltage change needed to be measured. The motor rotor was held locked until a set current level was reached, overcoming the initial effects of rotor inductance, and then released. Once again, the rotational velocity is initially zero and subsequently negligible when compared with the rate of rotational acceleration within the first instance of motor release. To calculate motor/system inertia Equation (4.21) was rearranged, taking into account negligible initial velocity to give:

$$J = \left(k_t i \right) / \ddot{\theta} \tag{5.9}$$

The rotational acceleration was initially in units of V/s, but was then multiplied by the speed constant to convert to units of rad/s^2. With a steady release current of 17.2 A, the resulting system inertia is as reported in Table 5.2.

5.4 SUMMARY

Values for parameters identified in the dynamic equations of motion describing FWD wheelchairs and the equations of the PD DC drive motors were obtained. The Parameters values were determined through a series of physical tests and relevant calculations. The parameter values are described, as well as the physical tests and calculations, and their final values are given. All of the initial parameter values were inherited from prior research. However, questions were raised during this research about the validity of several of the values. Hence, several physical tests and/or calculations were repeated.

In particular it was found that all six parameters for the PM DC drive motor equations needed to be adjusted. With regard to the parameter values associated with the dynamic motion equations, the CoG position was reconfirmed, the wheelchair mass was adjusted slightly for additional equipment introduced during this research and final mass moment of inertia (MMI) needed to be estimated. Very little information was available to access the validity of the original MMI value until recently. However, this research was able to confirm the inherited MMI value was inaccurate due to flaws in the relevant test performed. A reasonable estimate was made based on newly acquired information that allowed a

comparison between the MMI values of several different EPWs with similar masses and dimensions to the FWD wheelchair test-bed.

6

MODEL ANALYSIS

6.1 INTRODUCTION

The model analysis was carried out in two parts. First, the PM DC motor system model was compared to measurements of the actual motor drive system because this data was available first and it is the fundamental component of the wheelchair system. Second, data collected during live wheelchair tests was used in comparison to validate the full, complete wheelchair system model with particular focus on the model's steering and stability dynamics.

6.2 PM DC MOTOR SYSTEM MODEL ANALYSIS

6.2.1 Testing Process

The relevant measured outputs have been discussed in Section 5.3.2 for the speed constant, where steady state values of voltage input, motor current and output angular velocity of the motor shaft were recorded under no load conditions. In particular, the velocity of the rotating wheel was taken to be equal to the rotational velocity of the of the geared PM DC

motor shaft. This assumption is a result of 'lumping' the moment of inertia of the motor rotor with the gearing and wheel rim including the tire. Figure 6.1 shows the actual relevant steady state measurements over an arbitrary time period, hence the areas of no data between steady state values.

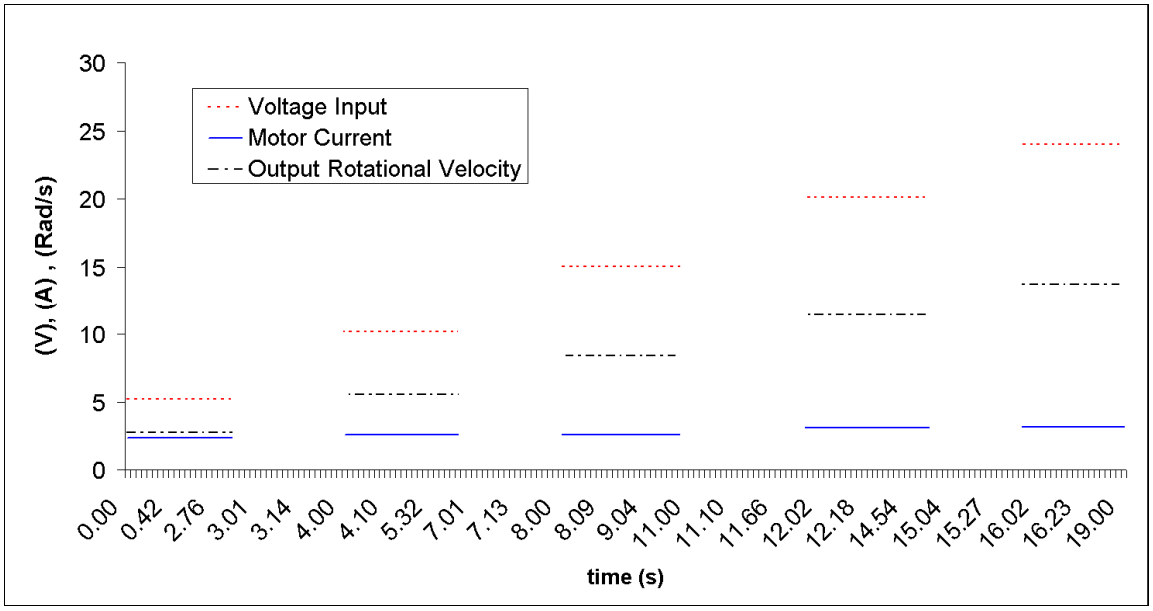

FIGURE 6.1: MEASURED STEADY STATE VALAUES OF PM DC DRIVE MOTOR SYSTEM FOR GIVEN VOLTAGE INPUTS (WOLM ET AL., 2008)

It can be seen that the output rotational velocity increases as expected, as does the motor current although very slightly, also as expected. This behaviour is predicted in Equation (4.22) where under higher rotational velocities the back emf, V_{emf}, becomes predominant. Hence, voltage is limited internally in the motor and therefore the motor current is also limited. Once a load is applied, back emf is decreased as the motor slows. This change increases the voltage difference internally thereby increasing current flow and the torque to maintain a given rotational velocity for a given voltage input and loading.

6.2.2 Results

The same steady state values of voltage were input to the PM DC motor model system. However, instead of duty cycle demands, the model was minimally adjusted to take straight voltage inputs. As stated in Section 4.3.6, for the state space model of an ideal PM DC motor, the output can be defined in terms of rotational velocity of the output shaft/rotor or the motor current. These variables were the two outputs of the model that were compared with measurements of the actual drive motor system, as seen in Figure 6.1. Figure 6.2 shows the measured current and the equivalent model output for the same given voltage inputs.

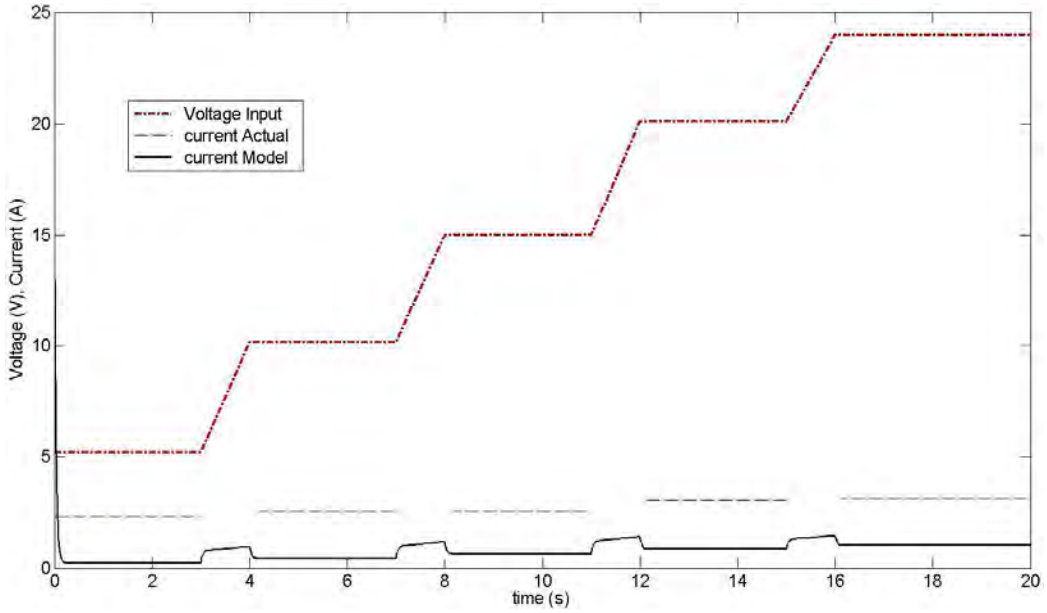

FIGURE 6.2: ACTUAL CURRENT MEASURED COMPARED WITH EQUIVALENT MODEL OUTPUT (WOLM ET AL., 2008)

The Simulink model has interpolated between the steady state voltage inputs to give the continuous graph of voltage. Furthermore, the Simulink model interpreted the initial voltage entered, which was not zero, as a step input. Hence, there is an initial current spike

that emulates what would actually happen in a PM DC motor when a sudden step input of voltage occurs and recorded continuously. Note that the subsequent current spikes are 'muted'. They would be similar to the initial current spike. The muting is due to the relatively shallow slope of the voltage rise between inputs due to Simulink's solver interpolation. However, the intent is accurate, as are the steady state values for the given parameters. Most notable is the large difference between the actual current measured and the theoretical model's equivalent output.

Figure 6.3 provides contrast between the measured rotational velocity and the equivalent output of the motor model for the same given input voltages. It is clear that the model output of rotational velocity is much closer to the actual measurements in Figure 6.3 for this variable and equivalent measurement. The initial increase and gradual levelling of the model's rotational velocity, as seen to the far left of Figure 6.3, is also in keeping with what would be seen of an actual PM DC motor with voltage step inputs. The subsequent changes are the same as for the subsequent changes given in the current comparison of Figure 6.2.

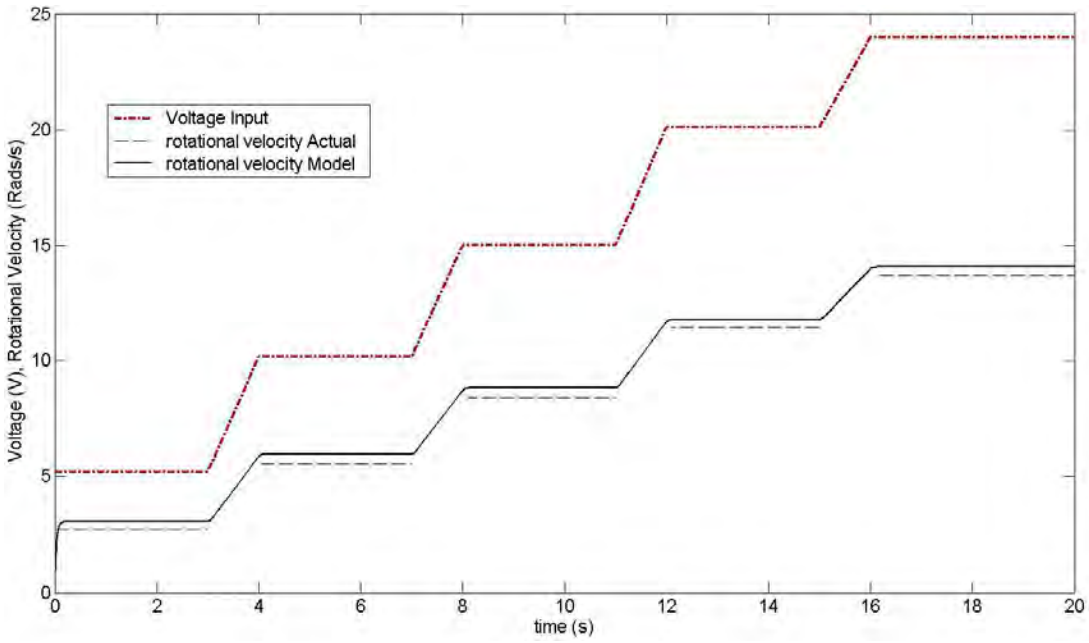

FIGURE 6.3: ACTUAL ROTATIONAL VELOCITY MEASURED COMPARED WITH EQUIVALENT MODEL OUTPUT (WOLM ET AL., 2008)

6.2.3 Analysis

The retesting of the of the PM DC motor parameters mentioned in Section 5.3, was due to these differences observed in Section 6.2.2. However, even with minor changes to the existing parameter values, there was no resolution of the differences in Figures 6.2 and 6.3. Hence, it can be ascertained that a specific dynamic or effect is missing.

As pointed out in the literature [29] there is a further parameter ignored by a host of academic papers and engineering texts. Others, such as Rizzoni [22], have made mention of a possible further parameter, but have down played its significance. Hence, the defining equations in the state space model presented in Section 4.3.6 represent an ideal PM DC motor system model. In any case, the particular missing parameter, known as Coulomb

friction, was discovered and applied during this research. The enhanced circuit including Coulomb friction can be seen in Figure 6.4.

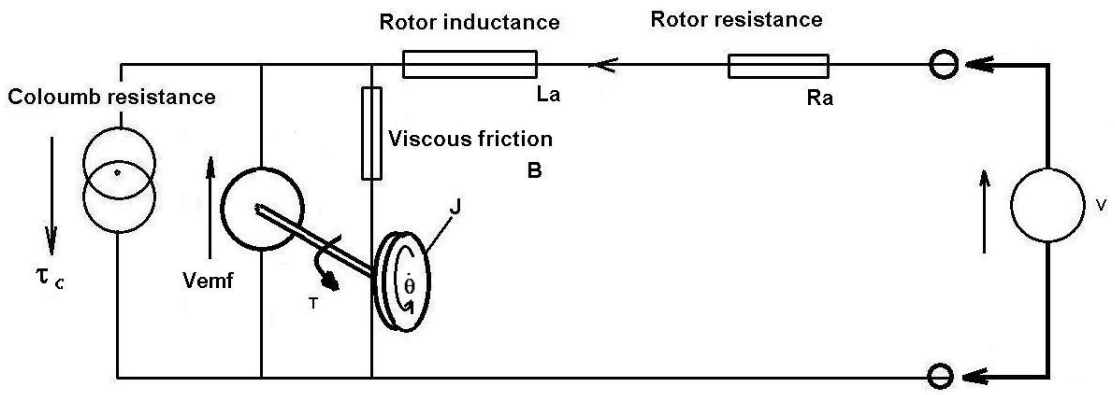

FIGURE 6.4: ENHANCED DC DRIVE MOTOR CIRCUIT (WOLM ET AL., 2008)

Coulomb friction is considered to be a constant retarding force, but is discontinuous over zero crossings. That is, when a motor reverses direction it must come to a stop, at which point Coulomb friction drops to zero, and then rises again in opposition to the reversed direction. In effect, Coulomb friction is constant when rotational velocity is not zero. In the case of modelling for feedback control of wheelchairs or other mobility vehicles, dynamic effects, such as over or under-steer, are noticeable under motion. In particular, the higher the velocities, the more noticeable are the effects.

In these cases, Coulomb friction can be treated as a linear constant. However, in this research it has been necessary to treat it as Heaviside constant in the model, so it does not still provide an output when the model's control inputs are zero, as it otherwise would. The model's only other concern is forward motion because reverse motion suffers from under-steer and not the destabilising effect that is of concern. Through the friction model, now recognising Coulomb friction, τ_c, Equation (4.21) is now defined:

$$k_t i = J\ddot{\theta} + B\dot{\theta} + H(\dot{\theta})\tau_c \tag{6.1}$$

where:

$H(\dot{\theta})$ = Heaviside function

The Heaviside function ensures that at this point τ_c opposes forward motion, but drops to zero when the wheelchair is stopped. The model was then rewritten graphically, as seen in Figure 6.5 without using the standard Simulink state-space block as it has no provision for a Heaviside function. The Heaviside function was instituted using a standard Matlab Embedded function block. It should be noted that the output is now $\ddot{\theta}$ and $\frac{di}{dt}$.

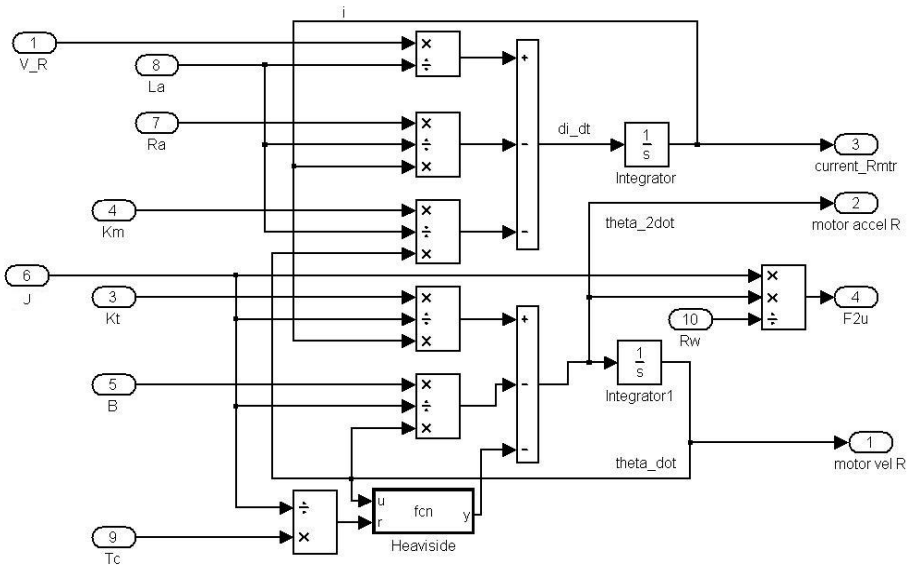

FIGURE 6.5: NEW PM DC MOTOR SYSTEM SIMULINK MODEL INCORPORATING T_C

Under steady state rotational velocity conditions and rearranging Equation (6.1) provides Coulomb friction in units of Nm,

$$\tau_c = k_t i - B\dot{\theta} \tag{6.2}$$

The same series of measurements as were used to calculate the speed constant were the values of input into Equation (6.2), namely current and rotational velocity. Both the torque constant and viscous friction coefficient had been calculated previously. The average of the series of calculations provided the Coulomb friction value of 4.0988 Nm.

Figures 6.6 and 6.7 compare the actual measured values for current and rotational velocity respectively with the equivalent model outputs for this model using the same given voltage inputs. As can be seen in both of the figures, the model outputs now more closely follow the actual measured values for current and rotational velocity. There is thus much less error compared to the equivalent ideal model plots of Figures 6.2 and 6.3. Table 6.1 provides the results of mean square error (MSE) method as a means to quantify the validation of the enhanced PM DC motor model using the steady state values found in Figures 6.6 and 6.7.

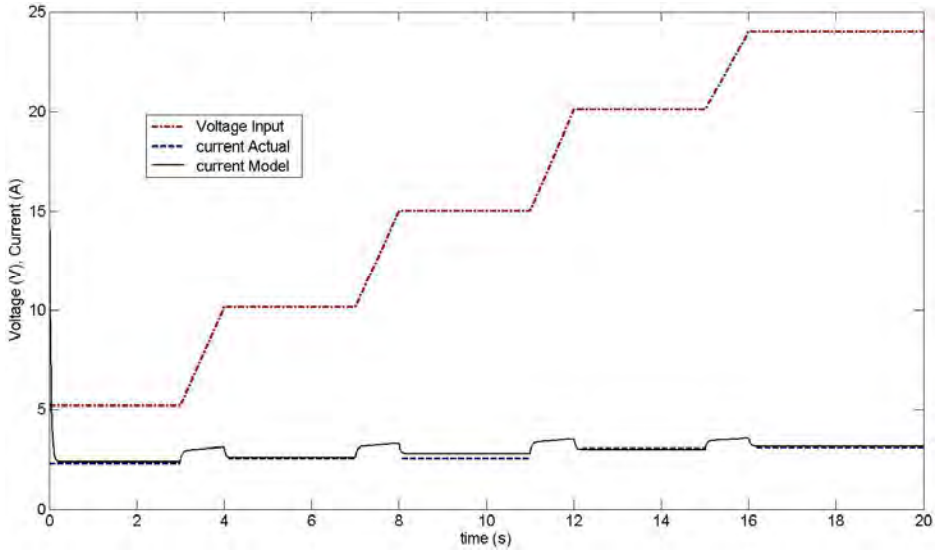

FIGURE 6.6: ACTUAL CURRENT MEASURED COMPARED WITH EQUIVALENT ENHANCED MODEL OUTPUT (WOLM ET AL., 2008)

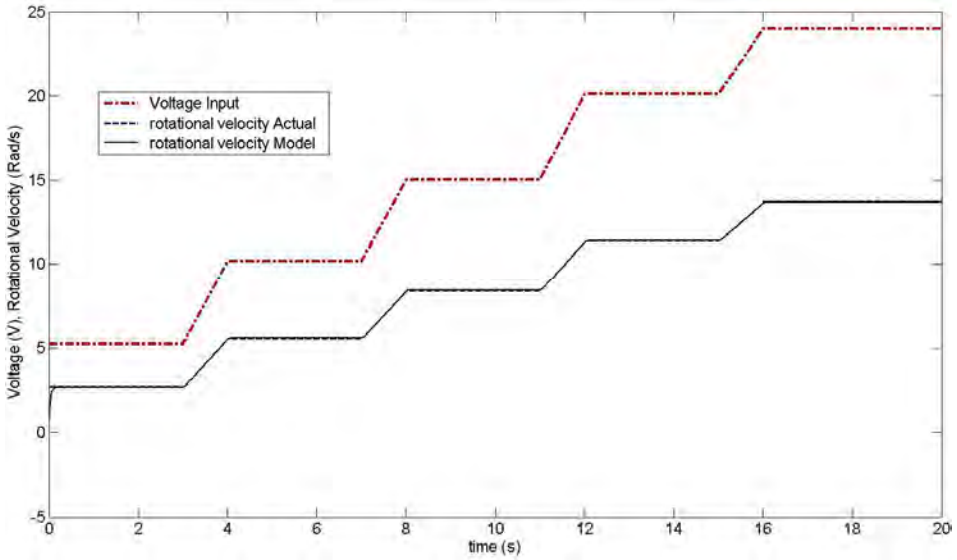

FIGURE 6.7: ACTUAL ROTATIONAL VELOCITY MEASURED COMPARED WITH EQUIVALENT ENHANCED MODEL OUTPUT (WOLM ET AL., 2008)

TABLE 6.1: MSE BETWEEN MOTOR MODEL AND ACTUAL MEASURED OUTPUTS

Voltage Input (V)	Rotational Velocity Error	Current Error
5.22	0.002	0.003
10.17	0.001	0.036
15.03	0.002	0.131
20.11	0.002	0.037
24.01	0.002	0.035

6.3 FWD WHEELCHAIR MODEL SYSTEM ANALYSIS

6.3.1 Testing Process

Data collected from patterns for an unoccupied wheelchair with all initial or main turns to the right were used to test the model's validity. This group of data, for 2 right doglegs, one

under control and the other with loss of control, and one right oval pattern, covered a broad spectrum and allowed the model to be tested under a number of different circumstances. The defined data group was also particular clean with no noise or other glitches.

Recall that in Section 3.2 the control duty cycle command inputs during the live tests were recorded and saved into a .mat file format. These particular .mat files were used for further live tests as also outlined in Section 3.2. Furthermore, they were also used in the validation procedure as it was important that the same inputs used on the live test-bed were the same inputs to the model. Hence, the recorded command inputs for the 2 right doglegs and right oval pattern were designated as the prime inputs for the testing process. Since the recorded duty cycle command inputs were in fixed steps, the model solver was required to be run with fixed steps. Hence, only ordinary differential equation (ode) solvers 1 to 5 were available in Simulink. Several simulations were run with a variety of the ode solvers with little or no noticeable differences in the solutions or solution times. However, all remaining simulations were run with ode5, the Dormand-Prince solver.

In Section 3.4.1 it states that the accelerometer data was to be used for model verification. However, as can be seen in Figures 6.8 and 6.9, comparing longitudinal and lateral accelerations with yaw respectively from a sample of one of the defined data group, that the yaw data is the much clearer and simpler of the data sets to use for validation.

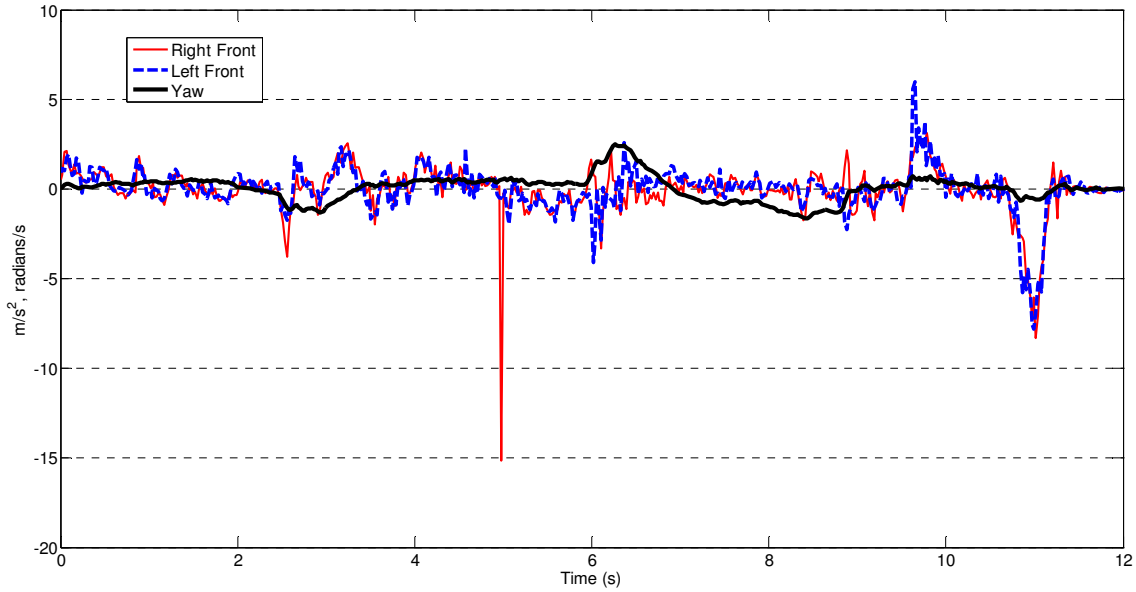

FIGURE 6.8: LATERAL ACCELERATIONS AT RIGHT AND LEFT DRIVE WHEELS VERSUS YAW AROUND VERTICAL AXIS

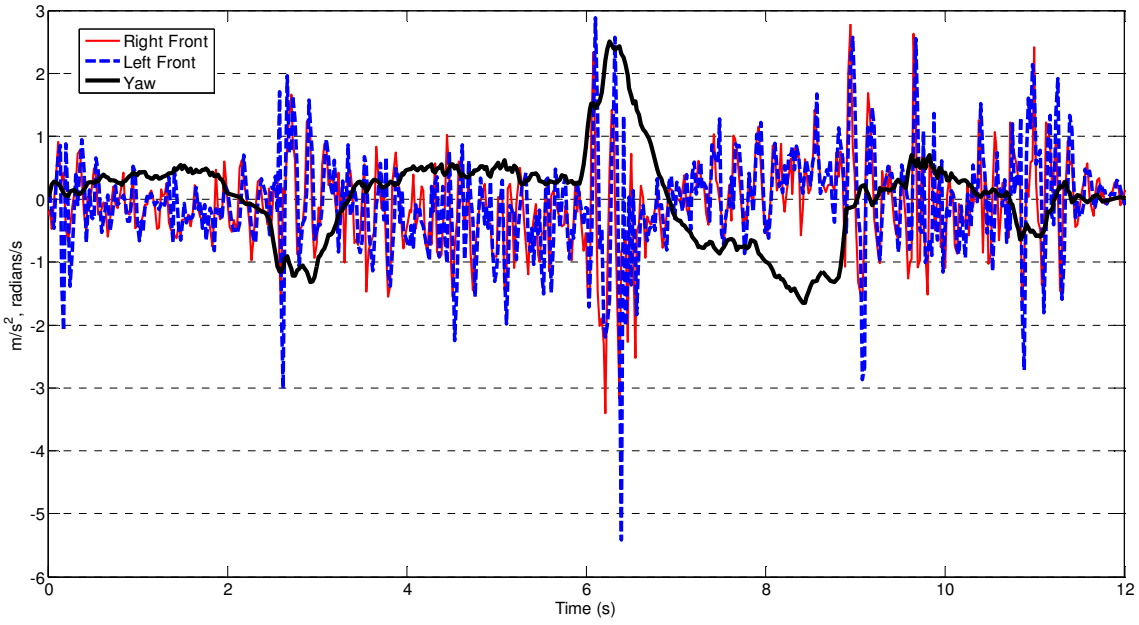

FIGURE 6.9: LONGITUDINAL ACCELERATIONS AT RIGHT AND LEFT DRIVE WHEELS VERSUS YAW AROUND VERTICAL AXIS

In addition, the yaw data had been fully processed previously, along with velocity data during the data analysis as also discussed in Chapter 3. Hence, yaw around the test-bed's vertical axis and forward velocity data were also used for model verification.

6.3.2 Results

Figures 6.10 to 6.15 compare the actual calculated velocity measurements of the designated test patterns with the velocity outputs of the model for the designated test patterns' inputs. However, Figures 6.10, 6.12 and 6.14 have an original mass moment of inertia, *I*, inherited from the previous research, while Figures 6.11, 6.13 and 6.15 have the new estimated value as reported in Table 5.2.

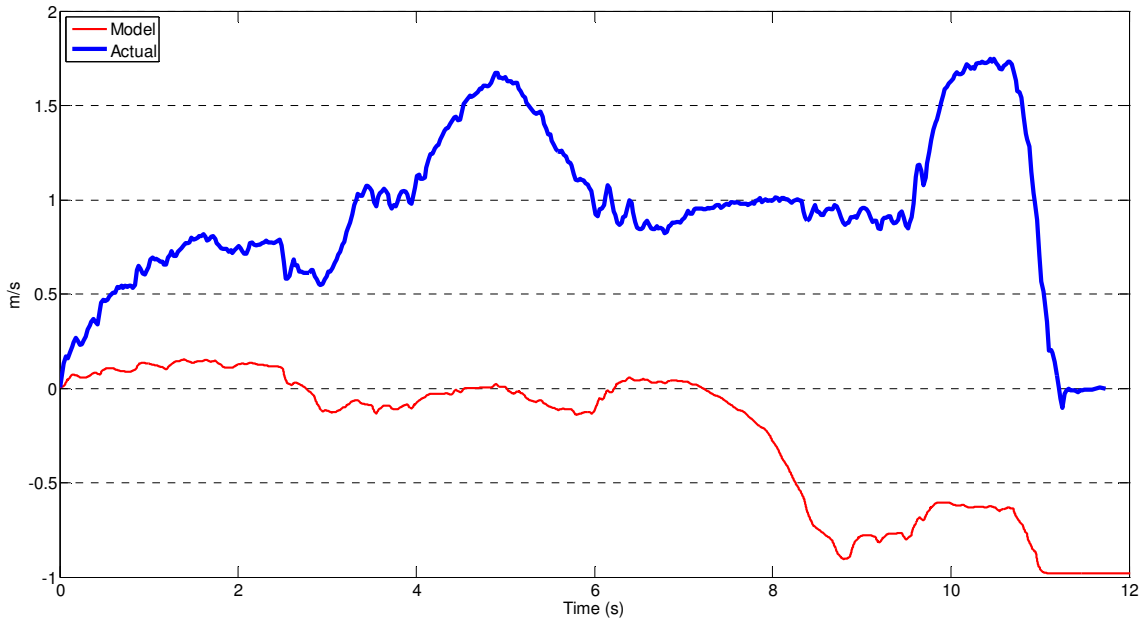

FIGURE 6.10: MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 0.871 KGM^2

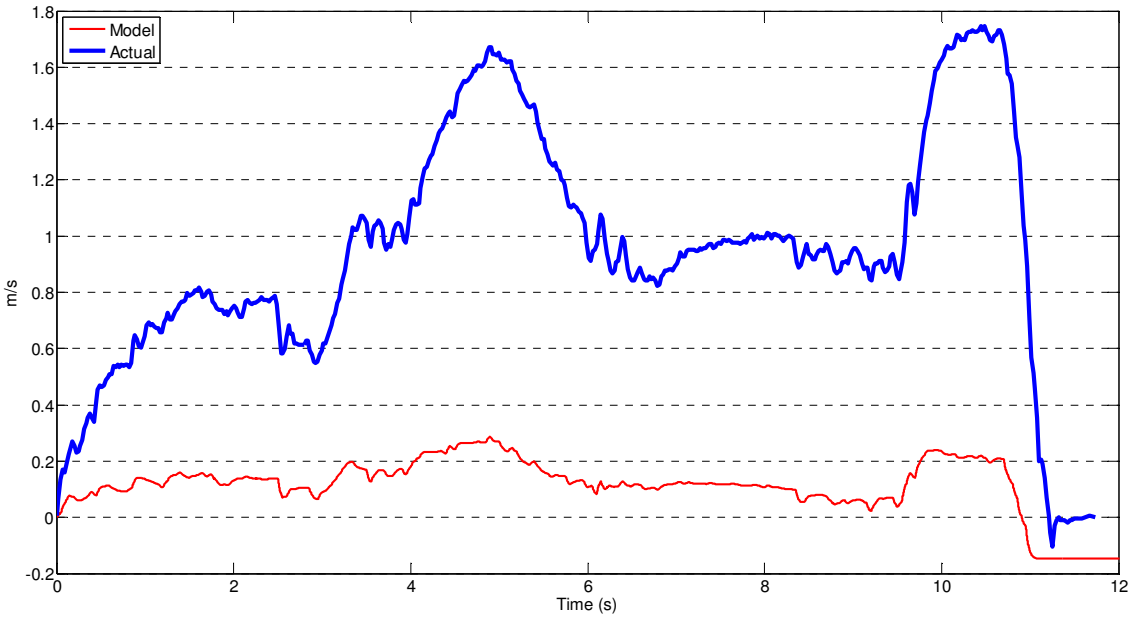

FIGURE 6.11: MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 5.5 KGM^2

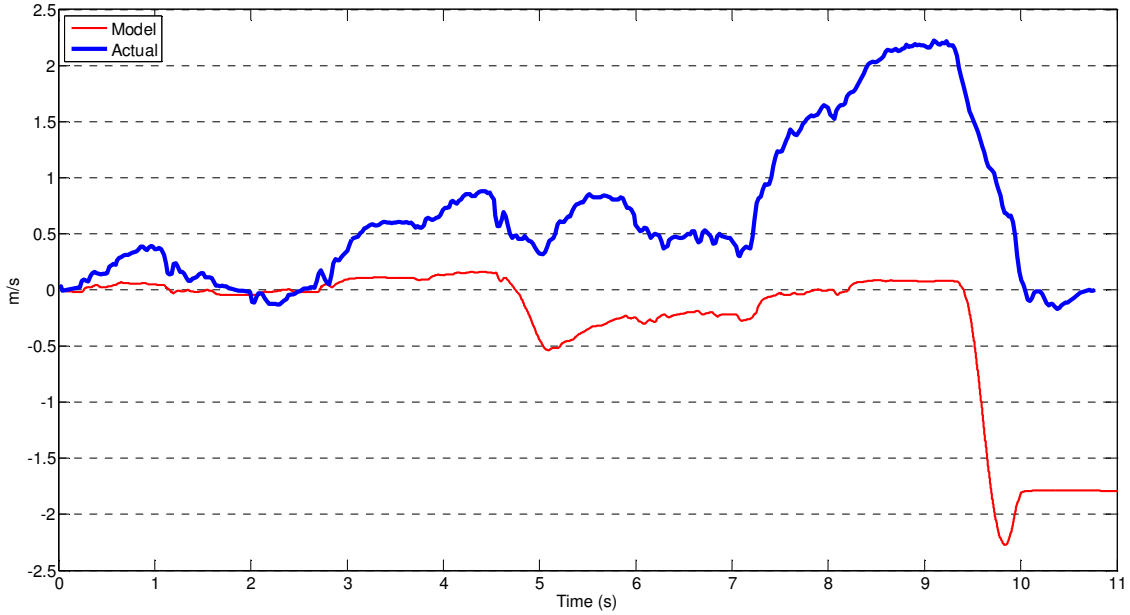

FIGURE 6.12: MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL WITH I = 0.871 KGM^2

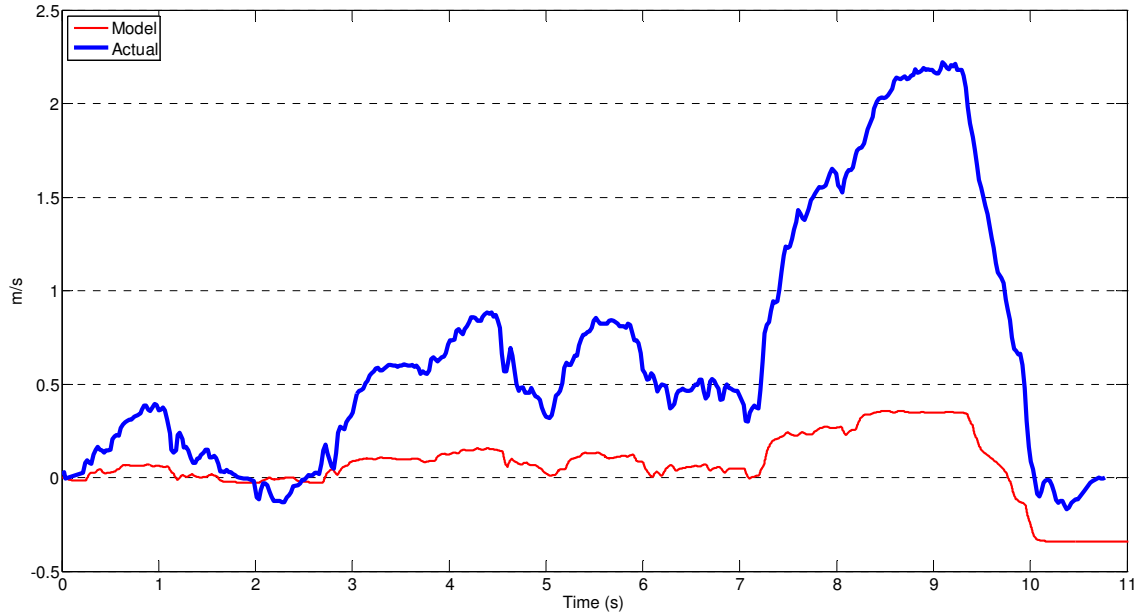

FIGURE 6.13: MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL WITH I = 5.5 KGM^2

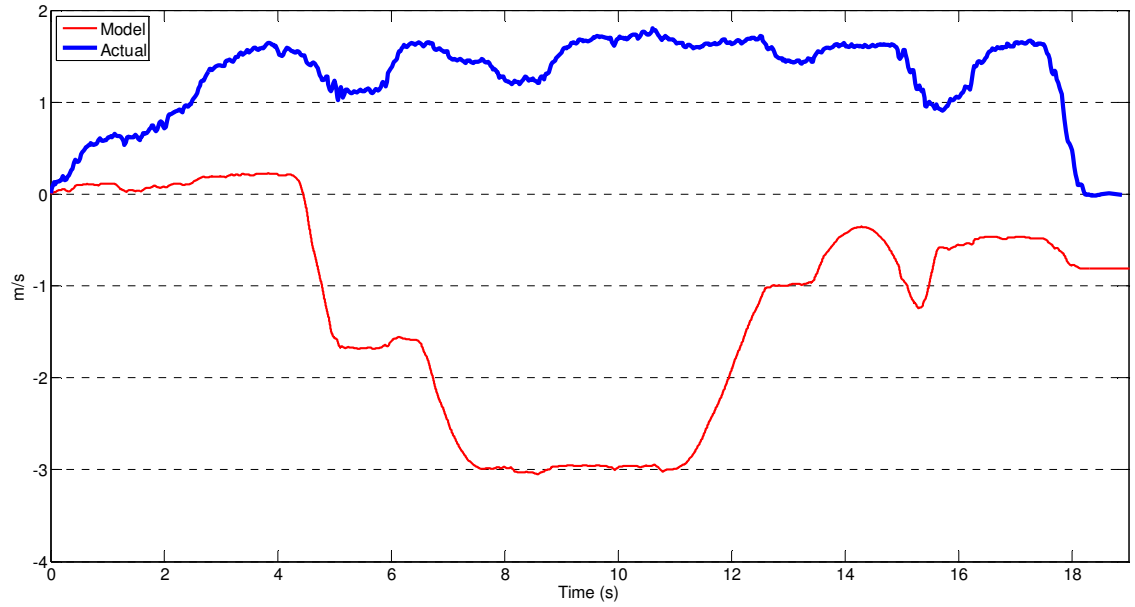

FIGURE 6.14: MODEL VS ACTUAL FORWARD VELOCITY FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 0.871 KGM^2

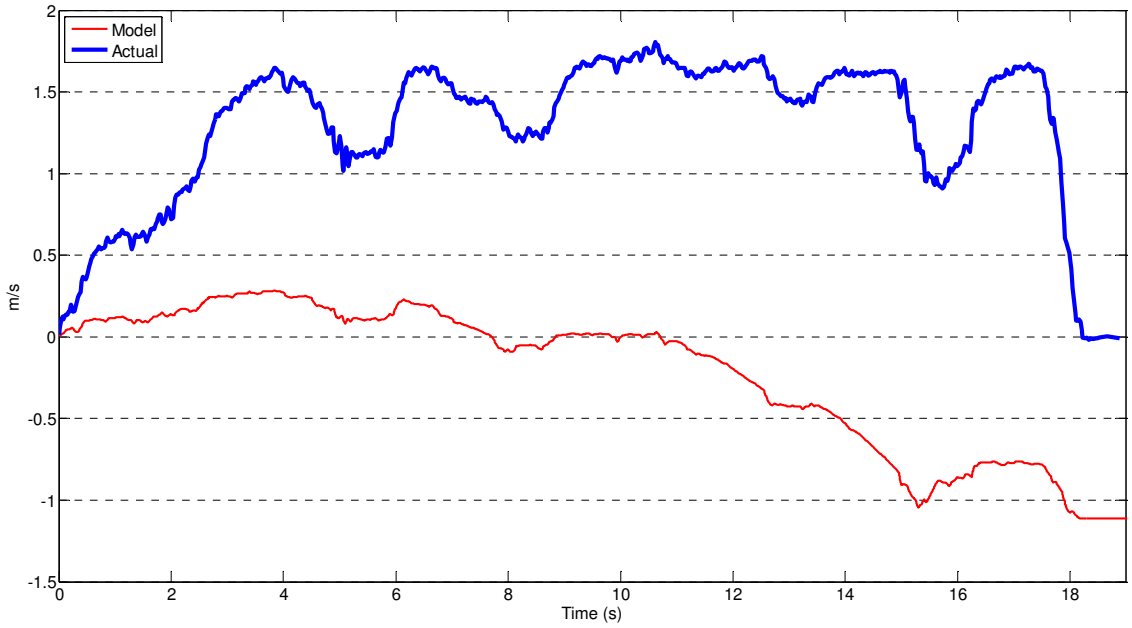

FIGURE 6.15: MODEL VS ACTUAL FORWARD VELOCITY FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 5.5 KGM^2

As can be seen in the figures comparing velocity, there is a wide disparity between the actual velocity and the model values for both settings of the mass moment of inertia. However, for those model's that have the new estimated value of inertia, the velocity curve contour from the model appears to mirror the actual velocity curve contour far more closely than those with the original mass moment of inertia.

Figures 6.16 to 6.21 compare the actual calculated rotational velocity or yaw measurements of the designated test patterns with the yaw outputs of the model for the designated test patterns' inputs. As with the velocity comparison figures, the even numbered figures in this series have the mass moment of inertia set to the original figure with the odd numbered figures containing the new estimated value.

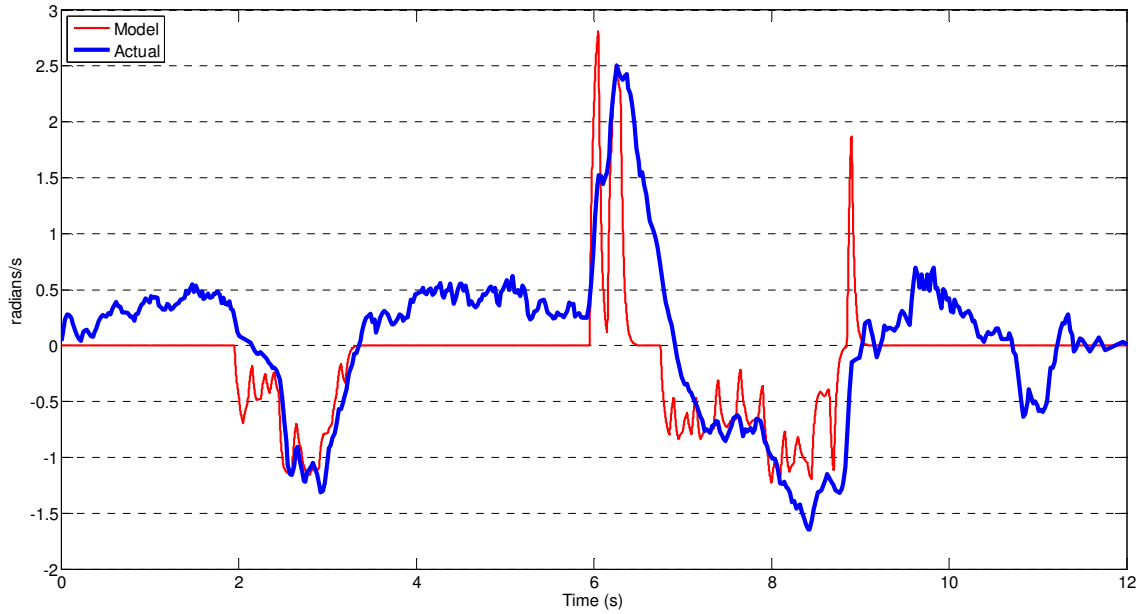

FIGURE 6.16: MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 0.871 KGM^2

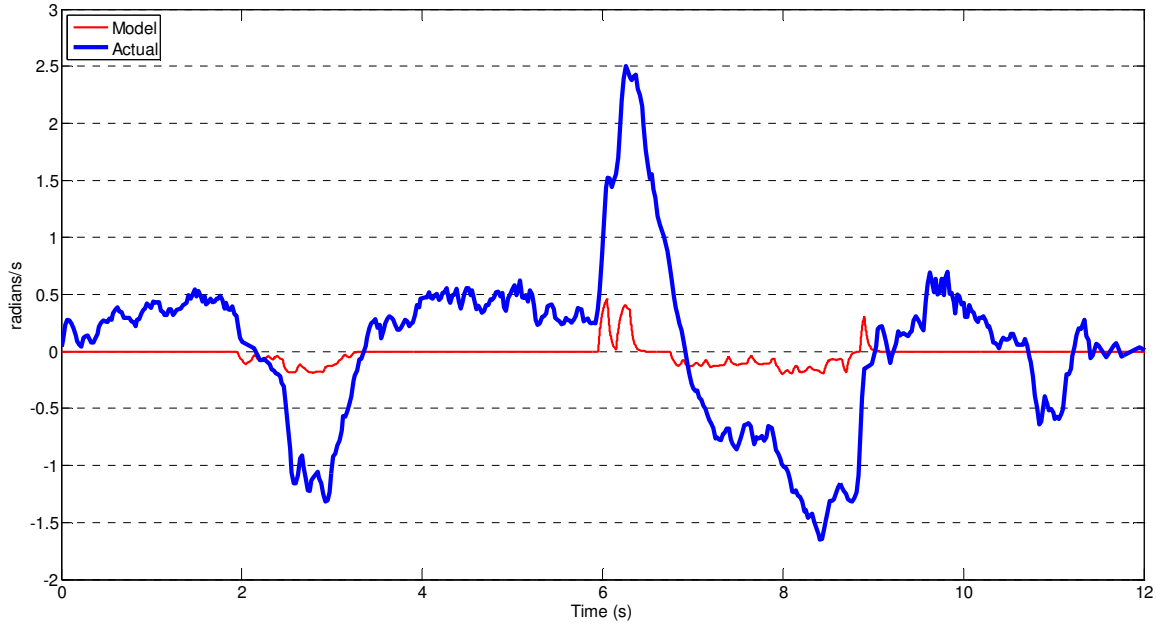

FIGURE 6.17: MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 5.5 KGM^2

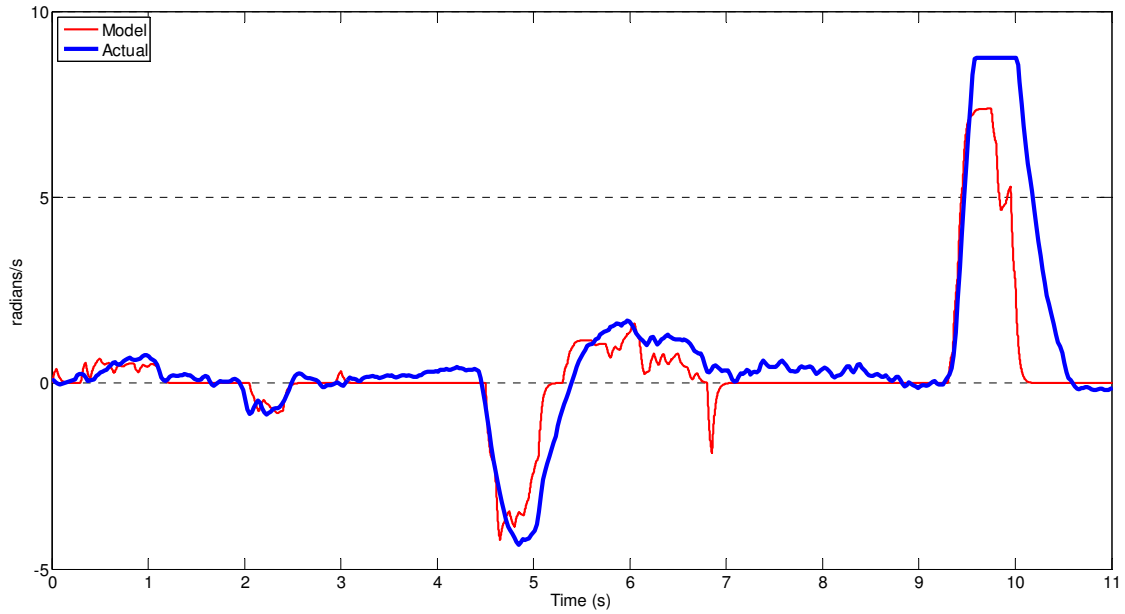

FIGURE 6.18: MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL WITH I = 0.871 KGM^2

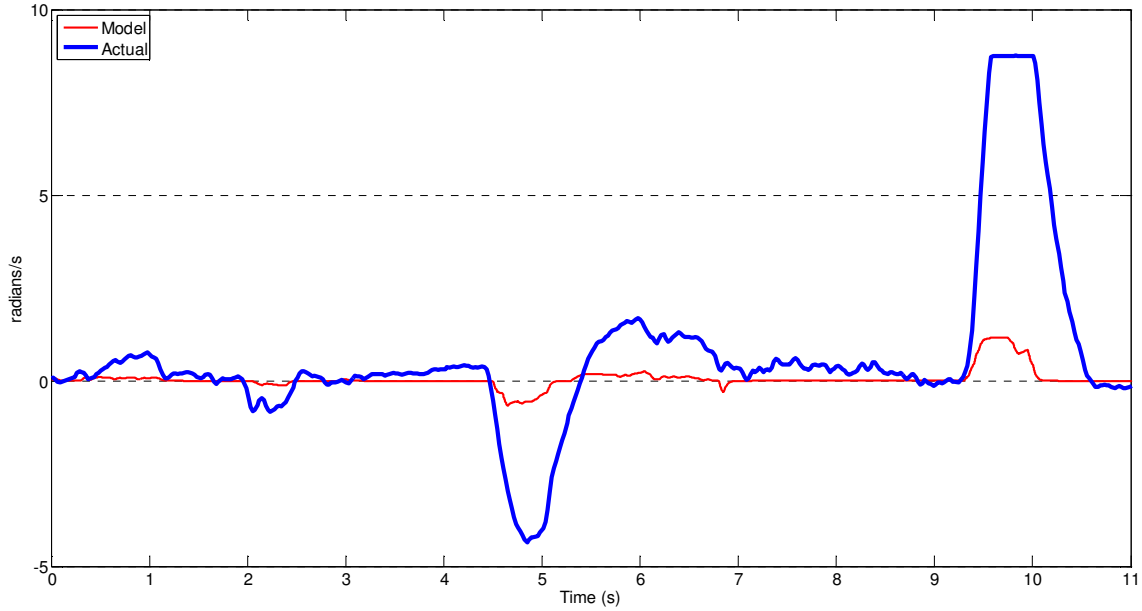

FIGURE 6.19: MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL WITH I = 5.5 KGM^2

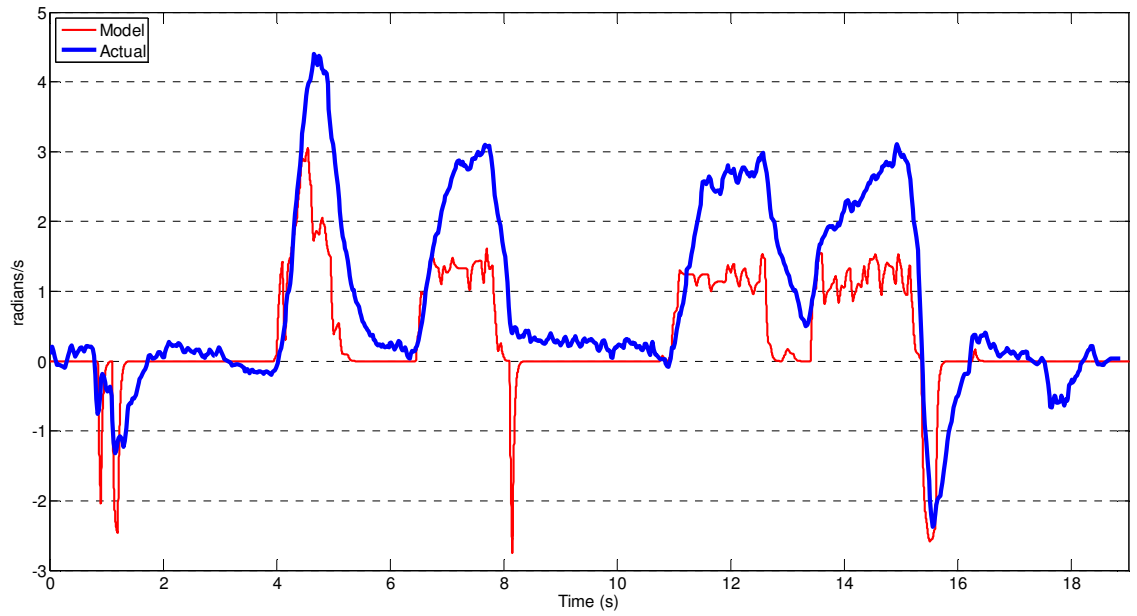

FIGURE 6.20: MODEL VS ACTUAL YAW FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 0.871 KGM^2

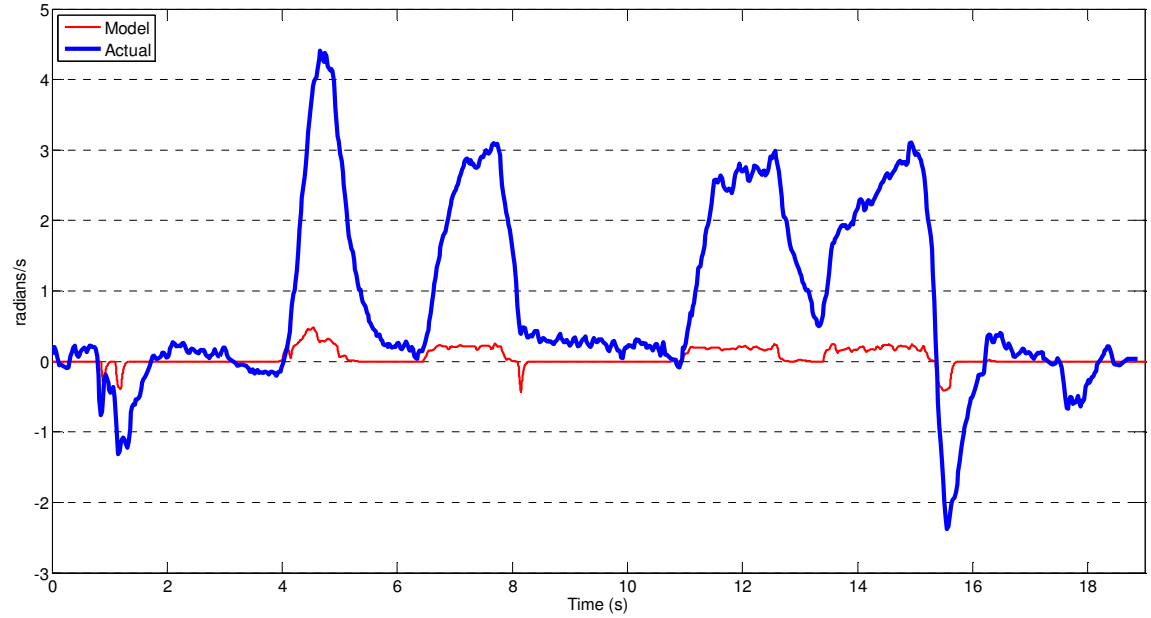

FIGURE 6.21: MODEL VS ACTUAL YAW FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL WITH I = 5.5 KGM^2

Unlike the curve contours for the velocity comparisons, there is no change in the curve contours for both values of inertia. However, with the higher inertia of the new estimated value, the difference between the actual and model yaw values has increased. This result stands to reason because as confirmed in Chapter 3, an increase in the mass moment of inertia means the rotational velocity rate and thus acceleration will decrease linearly.

6.3.3 Analysis

From Figures 6.10 to 6.21 it can be seen that having a more accurate moment of inertia improved the velocity curve contour, but, as reported, decreased the values of yaw for the model. From the results it appeared that there was an issue in scaling and indicated that a possible problem would be the voltage input to the model motors. Specifically, it was a question if the Roboteq motor controller on the wheelchair test-bed was actually supplying the expected voltages to each drive wheel.

According to Roboteq, the output of their motor controller should be linear. Hence, if the battery supply was 24 V and a control demand of 50% duty cycle was made, the supply to the motors should be 12 V, 24 V at 100% demand, and so on. There were certain static tests referred to in Section 3.1 to address this issue. The inputs for the no-load tests comprised a series of identical commands to each drive motor that increased the velocity of the drive wheels in steps from fully stopped to full forward rotational velocity. The results from the static tests were used to discover that the Roboteq motor controller was not providing the required voltage input to the drive motors in line with the duty cycle

command inputs linearly. A sample result of one of the tests can be seen in Figure 6.22 carried out with a battery supply voltage of 25.42 V.

The equation of the dashed fitted line in Figure 6.22 indicates that a 5th order Polynomial was needed. Furthermore, there is no voltage output from the controller until there is at least a 13% demand made and the output of the controller saturates at 90% demand which are more significant non-linearities. A Least-Squares approach was needed to create a model of the Roboteq motor controller as at different battery voltages, even though the order of the Polynomial fit equations remained the same, the Polynomial fit equations' coefficients changed. In addition, it is not fully known whether changes in input voltage (ΔV) less than 13% demand would also yield a zero change in output creating a far more non-linear system.

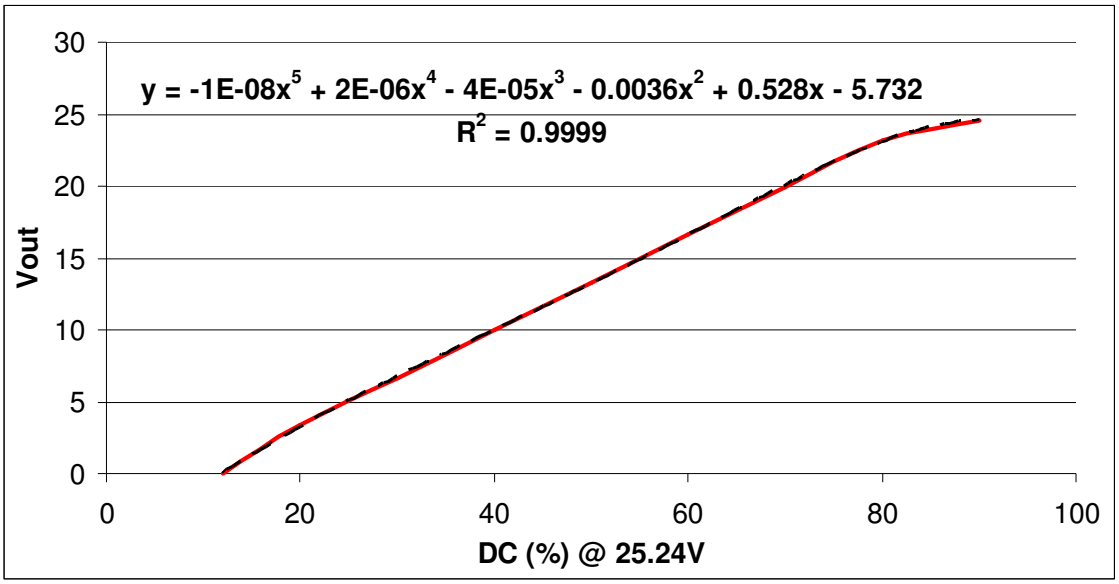

FIGURE 6.22: ROBOTEQ MOTOR CONTROLLER OUTPUT VOLTAGE IN RESPONSE TO GIVEN DUTY CYCLE DEMANDS

A successful model of the Roboteq motor controller was created but there was only a slight improvement to the scaling or curve contours of the wheelchair system model's velocity and yaw outputs. The lack of greater improvement necessitated further investigation of the dynamic equations and how they were applied in the model.

As mentioned in Section 1.4.2 the reactions of the caster wheel forces were to have been ignored, as their effects are deemed negligible in any direction once in motion. It can be seen in Figure 4.1, Chapter 4, that "alpha", which is actually $\delta_{1,2}$, and forces F3u and F4u have been set to zero to eliminate most of the caster reaction forces. Thus, Equations (4.4) to (4.6) become in effect:

$$ma_u = F_{1u} + F_{2u} \tag{6.3}$$

$$ma_w = F_{1w} + F_{2w} + F_{3w} + F_{4w} \tag{6.4}$$

$$I_z\ddot{\theta} = (F_{1u} - F_{2u})\frac{R_r}{2} - (F_{1w} + F_{2w})d + (F_{4w} + F_{3w})(L - d) \tag{6.5}$$

However, an immediate problem can be seen as the equations still do not reflect the fact that there is no lateral stabilising force from the rear casters. Forcing the side forces on the caster wheels, F_{3w} and F_{4w}, to zero to correct the situation yields:

$$ma_u = F_{1u} + F_{2u} \tag{6.6}$$

$$ma_w = F_{1w} + F_{2w} \tag{6.7}$$

$$I_z\ddot{\theta} = (F_{1u} - F_{2u})\frac{R_r}{2} - (F_{1w} + F_{2w})d \tag{6.8}$$

From Equations (6.7) and (6.8) it becomes apparent that the calculation using statics and taking moments around various axes for the side forces on the drive and the caster wheels, as reported in Section 4.3.3, was based on a false premise.

With the centripetal force, F_c, acting on a wheelchair's CoG in a turn, the only opposing side forces can be from the drive wheels as confirmed by Equation (6.7). In essence, with no side slip or skid in a turn a static situation exists in lateral direction. Thus, relating the centripetal force to the side forces on the drive wheels and referring to Equations (4.14) and (4.15) yields:

$$F_c = F_{1w} + F_{2w} \quad \underline{\text{OR}} \quad mv_u\dot{\theta} = F_{1w} + F_{2w} \tag{6.9}$$

The changes to the "Main Plant Dynamics" block entailed by Equations (6.6) to (6.9) can be seen in Figure 6.23.

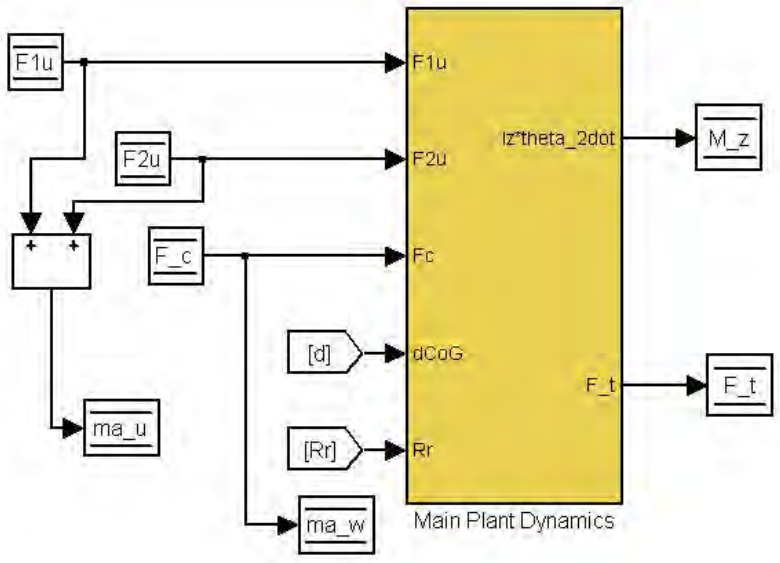

FIGURE 6.23: NEW MAIN PLANT DYNAMICS BLOCK IN DYMANIC FWD SYSTEM MODEL

The changes to model yielded improvements in the velocity values, but further decreases in the yaw values. However, the curve contours of both the velocity and yaw results are maintained or improved. Figures 6.24 and 6.25 show the results of the changes for one set of the defined data group.

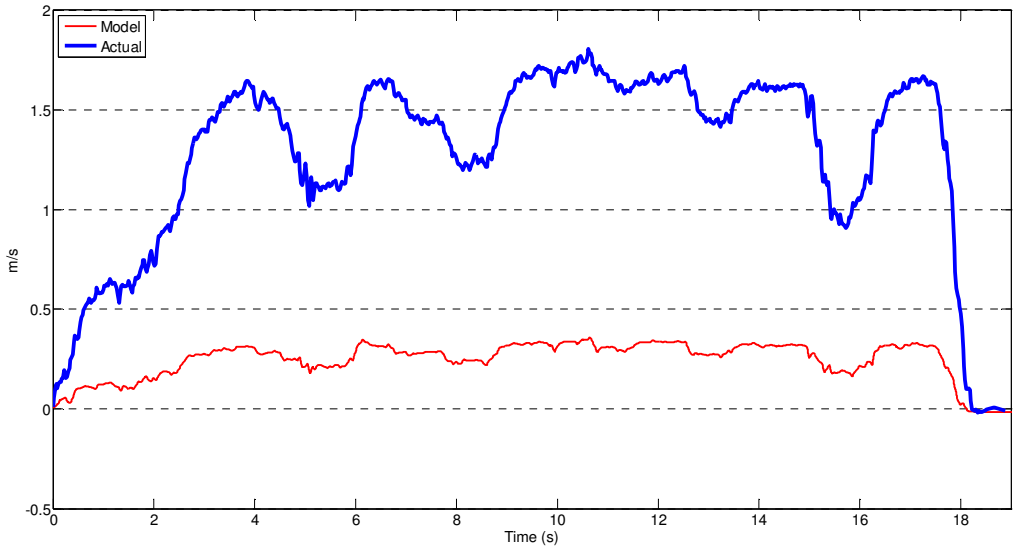

FIGURE 6.24: CHANGED MODEL VS ACTUAL FORWARD VELOCITY FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL

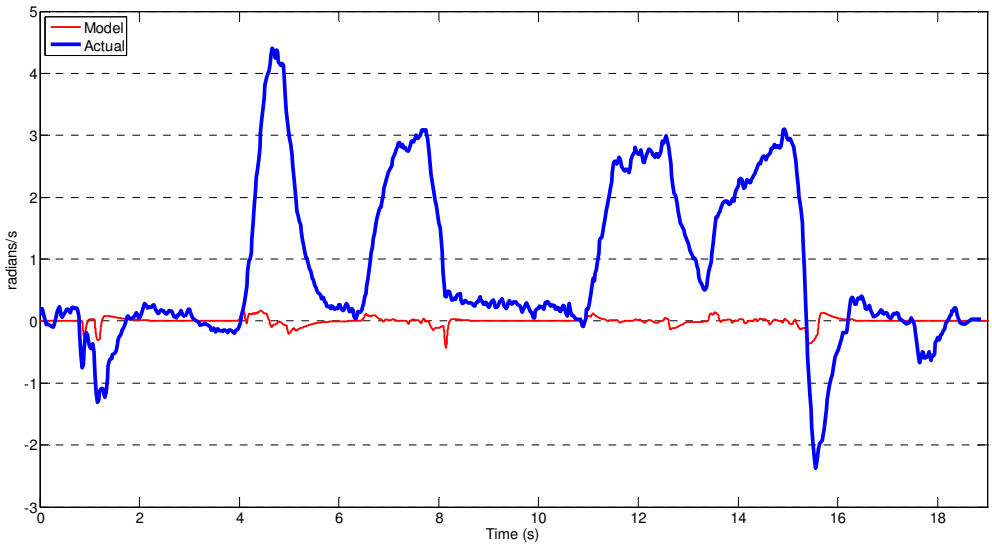

FIGURE 6.25: CHANGED MODEL VS ACTUAL YAW FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL

Further investigation revealed that Equation (4.15) used in the model to determine centripetal force has an incorrect variable. As described in Chapter 4, Section 4.3.3, the rotational velocity, $\dot{\theta}$, replaced the radius of curvature term, *r*, as $\dot{\theta}$ was readily available in the model. However, the rotational velocity available in the model was rotational velocity around the wheelchair's vertical axis through the CoG, $\ddot{\theta}_z$, and not the required rotational velocity of the wheelchair around its instantaneous centre of curvature or ICC, $\dot{\theta}_{ICC}$. Figure 6.26 shows the ICC or ICR, defined in the figure, for a 4 wheeled differential drive robot with 2 drive wheels and 2 floating caster wheels according to Siegwart and Nourbakhsh [23]. It is clear that the ICC for this robot is applicable to a FWD wheelchair.

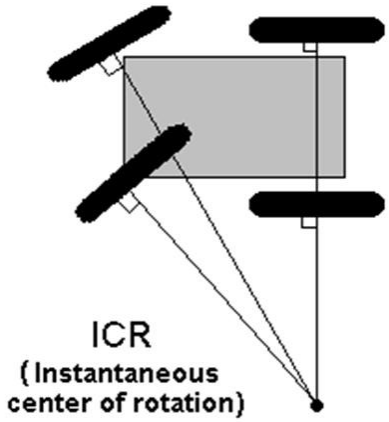

FIGURE 6.26: ICC FOR DIFFERENTIAL DRIVE MOBILE ROBOT AND A FWD WHEELCHAIR (SIEGWART AND NOURBAKHSH, 2004)

Figure 6.27 shows *R* or *r* in the current nomenclature, based on the ICC from which Han et al. [10] provide an equation for radius of curvature that is defined:

$$R = \left(\frac{v_R + v_L}{v_R - v_L}\right)\frac{l}{2} \tag{6.10}$$

where $l = R_r$ under this research's current nomenclature, the distance between the 2 driving wheels. In the model, the values for right and left drive wheel forward velocity, v_R and v_L, have been replaced in Equation (6.10) with ω_R and ω_L, the drive wheel rotational velocities to yield:

$$r = \left(\frac{\omega_R + \omega_L}{\omega_R - \omega_L} \right) \frac{R_r}{2} \tag{6.11}$$

as v_R and v_L are equivalent to $\omega_R r_{wheel}$ and $\omega_L r_{wheel}$, where r_{wheel} is the radius of the drive wheels. It can be seen that the wheel radius terms cancel resulting in Equation (6.11).

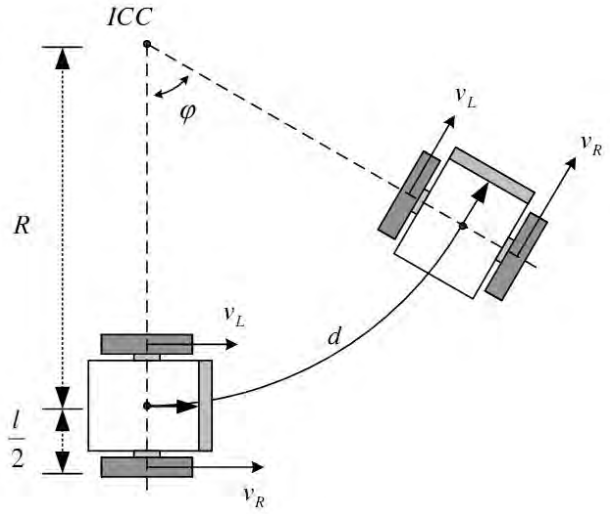

FIGURE 6.27: FINDING R FROM THE ICC FOR A DIFFERENTIAL DRIVE ROBOT OR FWD WHEELCHAIR (HAN ET AL., 2008)

Equation (6.11) was incorporated into the model such that equation (4.13) that defines centripetal force, F_c, with r was reinstated. The results of this alteration for one set of the defined data group can be seen in Figures 6.28 and 6.29.

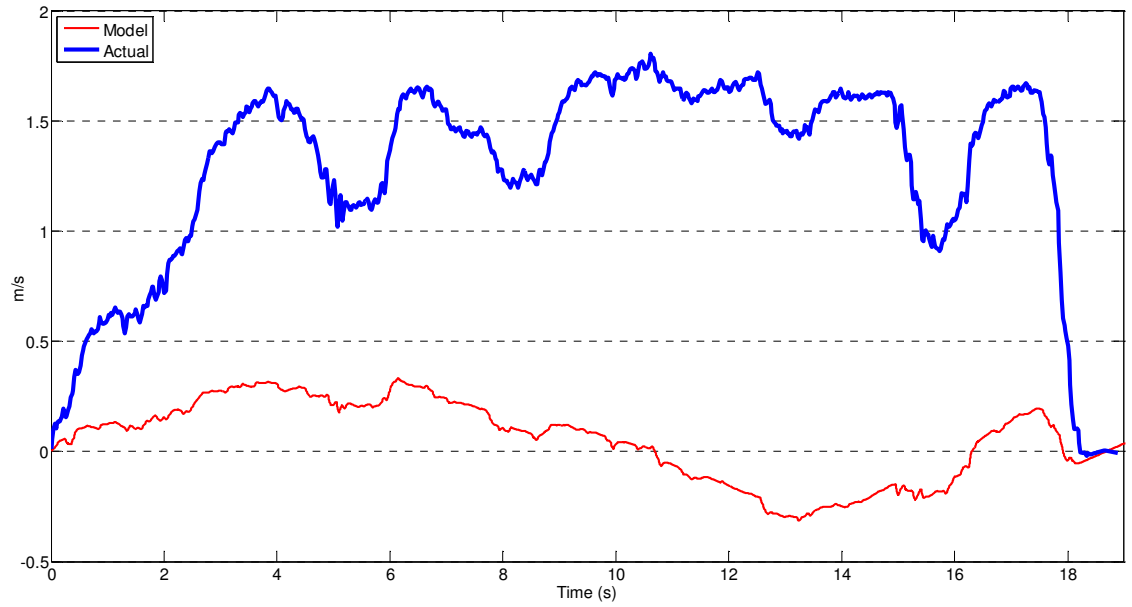

FIGURE 6.28: ALTERED MODEL VS ACTUAL FORWARD VELOCITY FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL

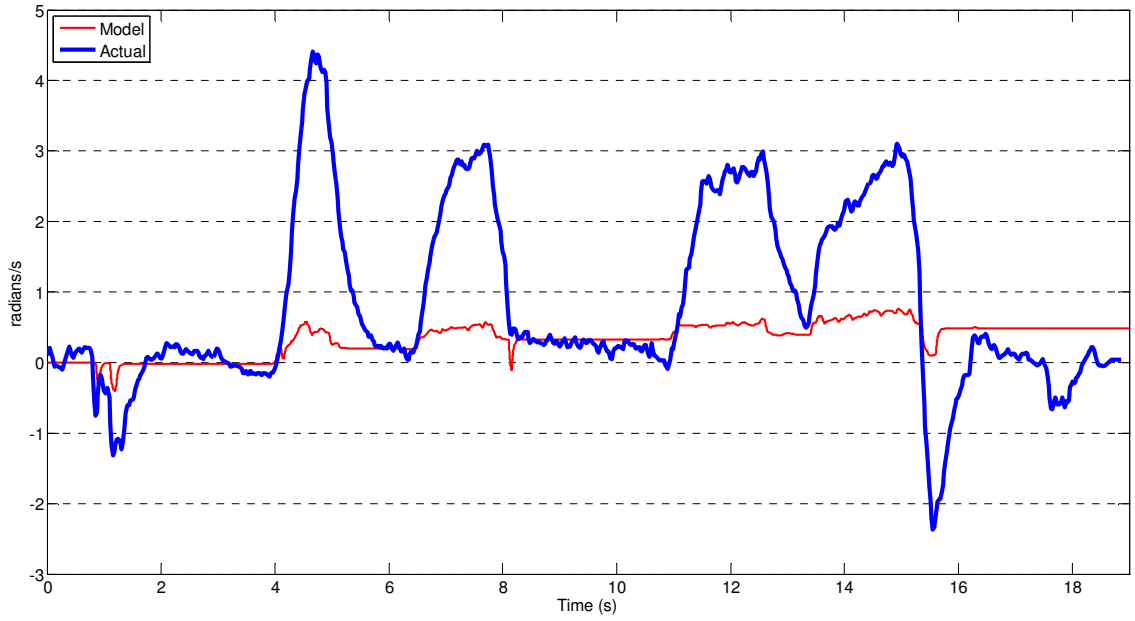

FIGURE 6.29: ALTERED MODEL VS ACTUAL YAW FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL

As can be seen the model's velocity values were degraded as the yaw values improved with all formulas now with the correct variables and parameters. However, according to [10], Equation (6.11) is an ideal one. This equation is used in trajectory planning of a mobile robot attempting to intercept a moving object [10], particularly following curved paths, but being an ideal equation has resulted in significant tracking errors. Han et al. have concluded that this error increases with higher velocities and tighter turns up to 100%. Hence, their stated solution for the centripetal force is defined:

$$F_c = c\frac{mv_u^2}{r} \tag{6.12}$$

where c is an undefined constant of proportionality. It appears that this sort of trajectory planning for mobile rear caster-wheel robots does not take into account over-steer and the solution is to drive small radii at very low forward velocities or turn in large radii at higher velocities, both of which minimise the over-steer effect. Hence, r, is greatly effected by over-steer, which did not suit this research's model as the radius of curvature must be independent of the effects of over-steer as is used to calculate over-steer itself.

As no reliable means could be found to calculate a viable radius of curvature value, a pure kinematic model was investigated that instituted Equations (4.1) to (4.3). This kinematic model can be seen in Figure 6.30. Figures 6.31 to 6.36 compare velocity and yaw values of the Kinematic model to the actual test-bed results for the defined data group reported in Section 6.3.1.

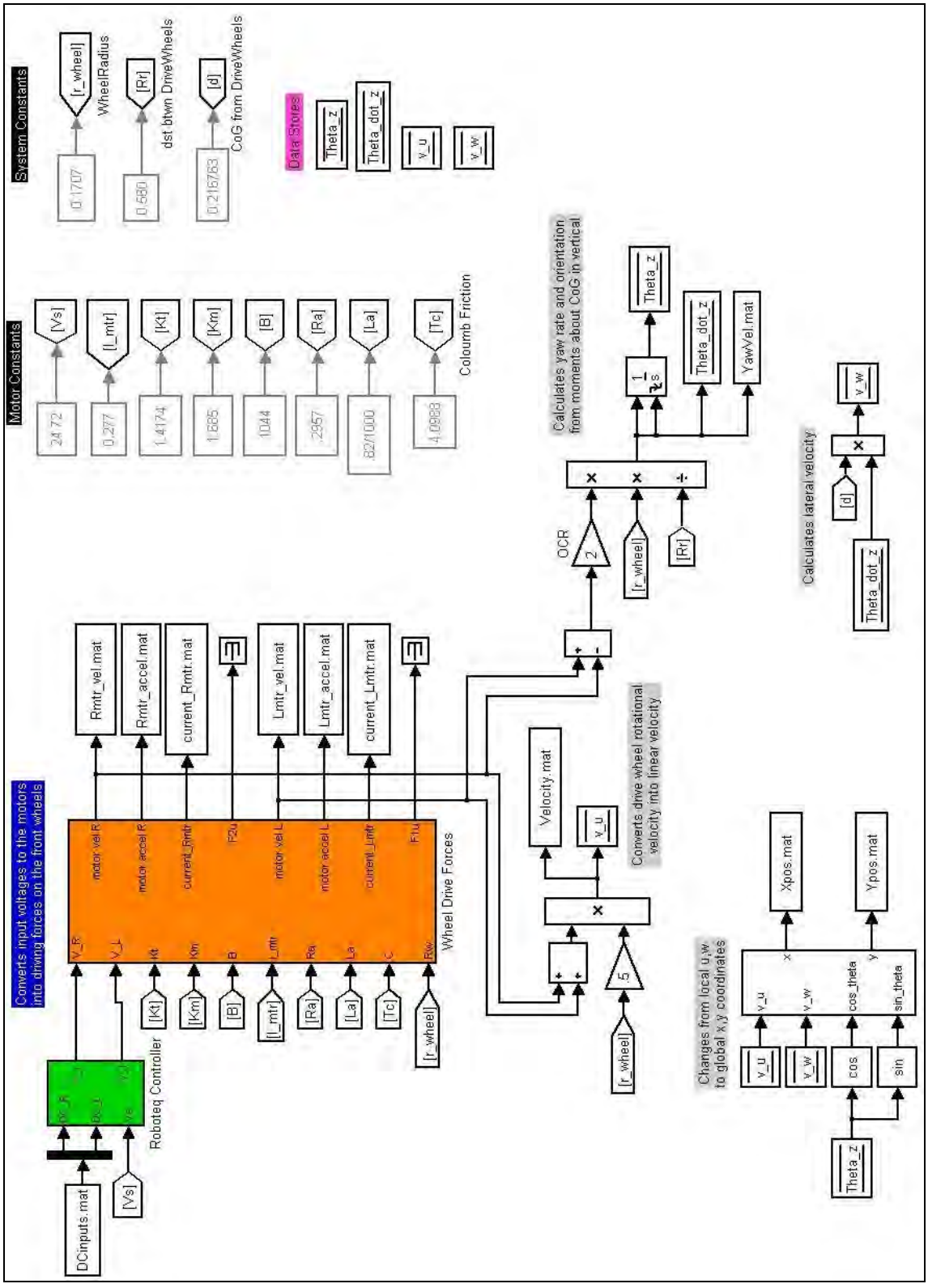

FIGURE 6.30: KINEMATIC FWD WHEELCHAIR SYSTEM MODEL

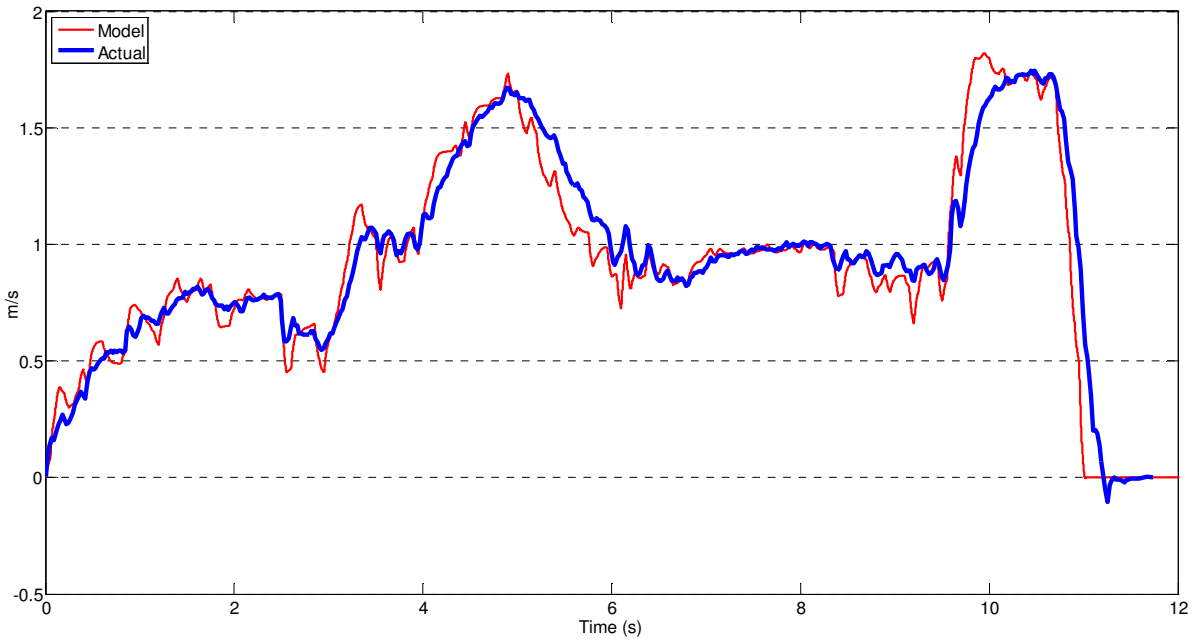

FIGURE 6.31: KINEMATIC MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL

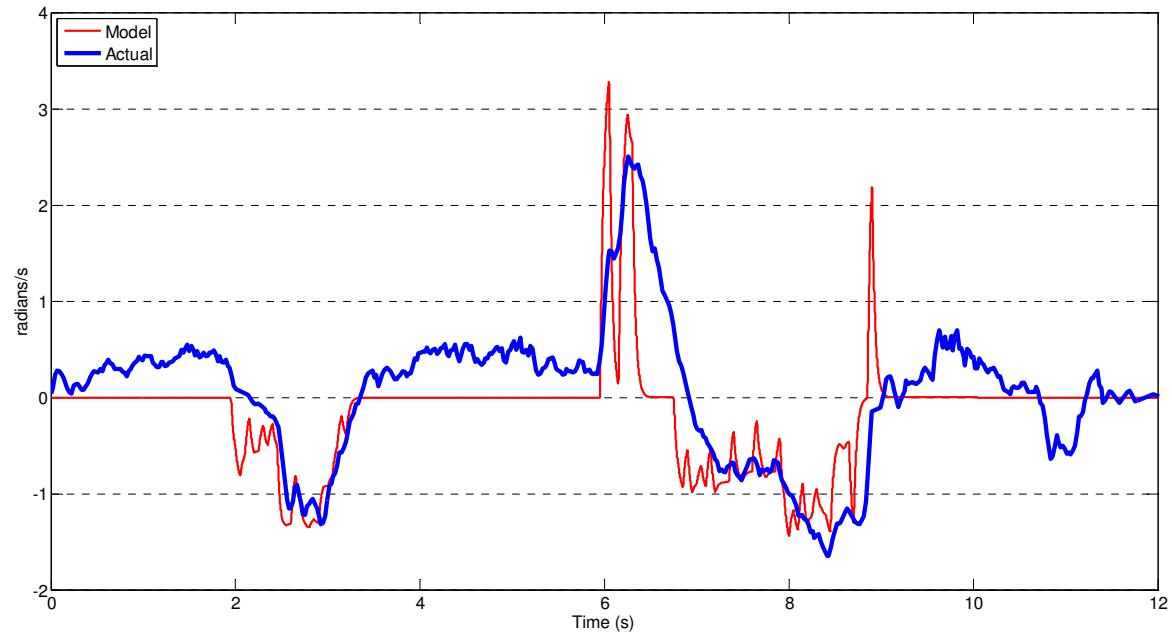

FIGURE 6.32: KINEMATIC MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL

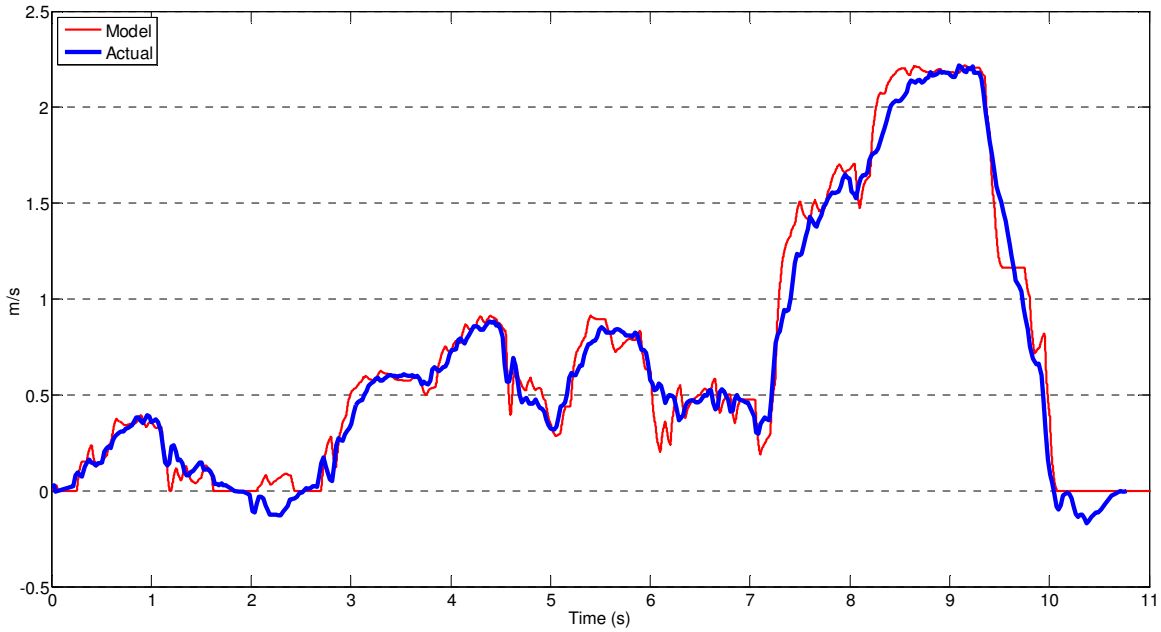

FIGURE 6.33: KINEMATIC MODEL VS ACTUAL FORWARD VELOCITY FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL

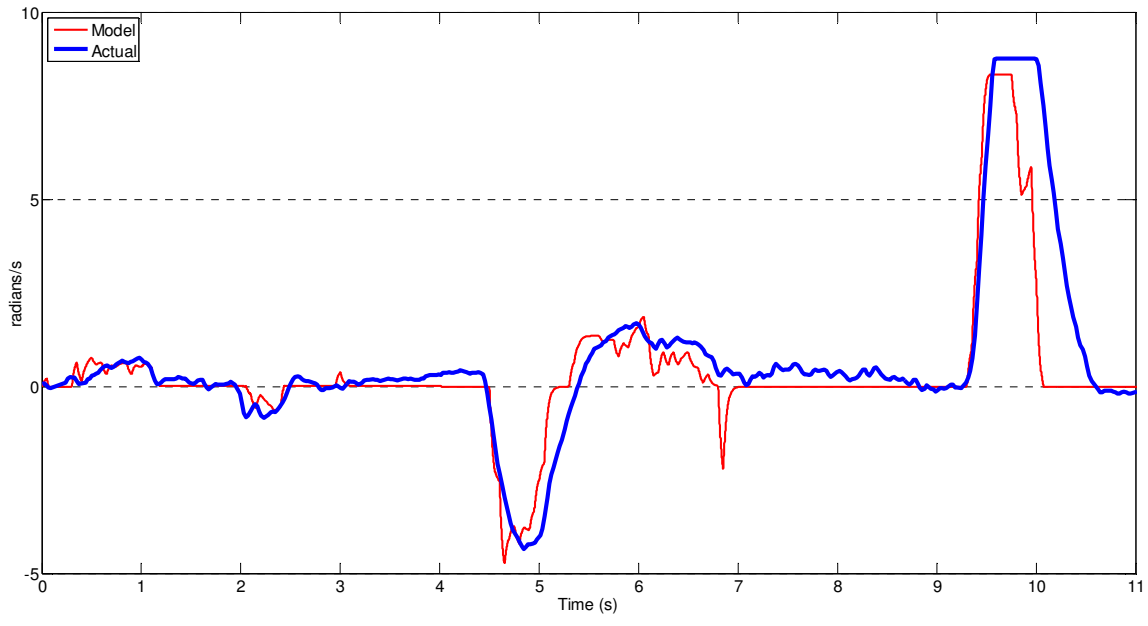

FIGURE 6.34: KINEMATIC MODEL VS ACTUAL YAW FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL

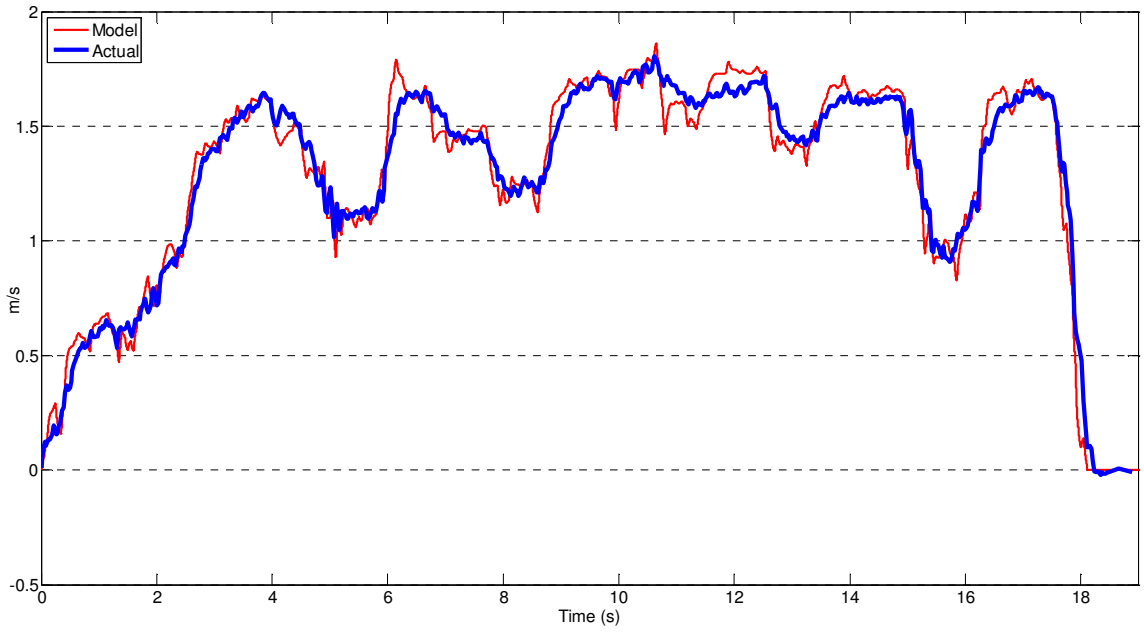

FIGURE 6.35: KINEMATIC MODEL VS ACTUAL FORWARD VELOCITY FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL

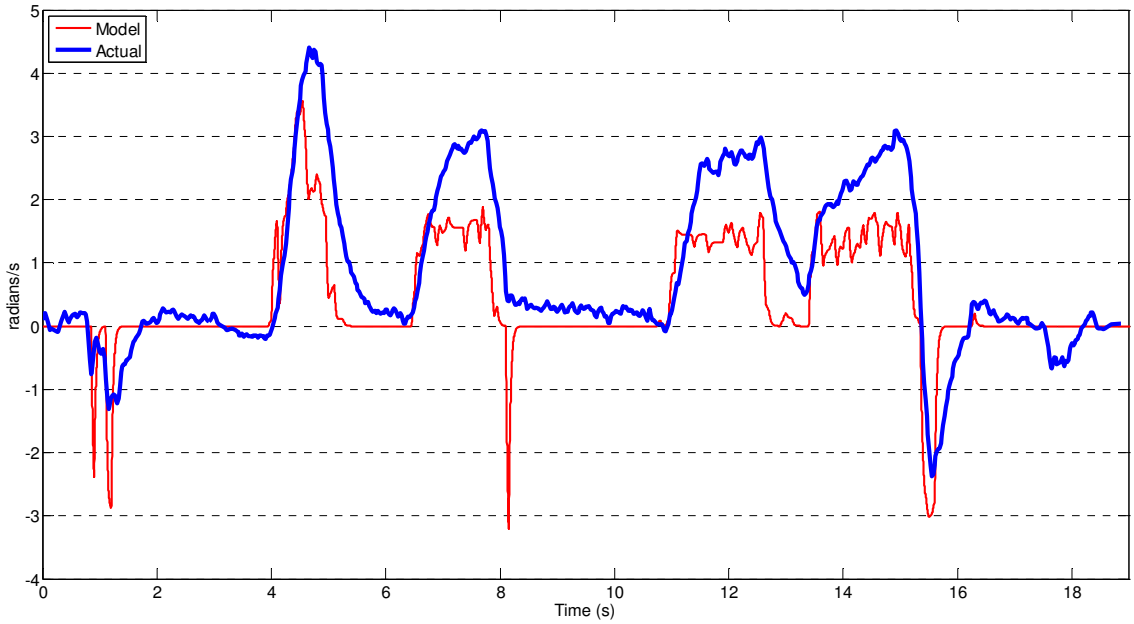

FIGURE 6.36: KINEMATIC MODEL VS ACTUAL YAW FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL

The velocity curve contours and values are virtually identical for the model and actual outputs in Figures 6.31, 6.33, and 6.35 particularly when compared with previous models' results. The yaw curve contours have changed little but the values have improved dramatically and are virtually identical to the traces of the same data group for the Dynamic model with the inaccurate moment of inertia value seen in Figures 6.16, 6.18 and 6.20.

Differences between the Kinematic model and actual velocity profiles can be attributed to lack of counter torque. There is no contribution for the counter torque caused by the road load in the kinematic model. However, these slight differences indicate that road load is relatively small. Furthermore, the model reacts to the inputs precisely whereas during the actual tests there is noise attributable to slight unevenness and changes in the texture of the road surface such as stones or loose dirt. This noise creates differences between actual and model values for both velocity and yaw.

For instance, in regards to yaw, in Figure 6.36 between the 8.2 and 10.85 second mark, model yaw is zero, but the actual yaw profile is showing a series of very tiny rotations or turns clockwise. However, the recorded demand inputs for this time period indicate that the wheelchair test-bed should have been driving straight forward. It is further possible that the wheelchair was turning because there may have been a slight delay between when each motor receives its voltage input due to the motor controller or wireless transmission link. Finally, the PM DC motor model has been based on the measurements of only one of the drive motors but applied to both motors in the FWD system model so any differences there can also contribute to the noise.

On closer inspection of the yaw output of the Kinematic model, a correction factor termed the "Over-steer Correction Factor" or OCF was included and proved to bring greater correlation between the actual and model yaw values. The OCF is justified on the grounds that the kinematic model by its very nature is not concerned with centripetal force and has been added to cater for this. However, the value of OCF = 2 was reached empirically and is thus not fully justified but is the most accurate value at present. Figures 6.37 to 6.39 show the drive motor wheels' rotational velocities that are integral to the calculation of the yaw rates. In particular, the various 'step inputs' in the wheel velocities that correspond to the spikes in the yaw rates in Figures 6.32, 6.34 and 6.36 are emphasised.

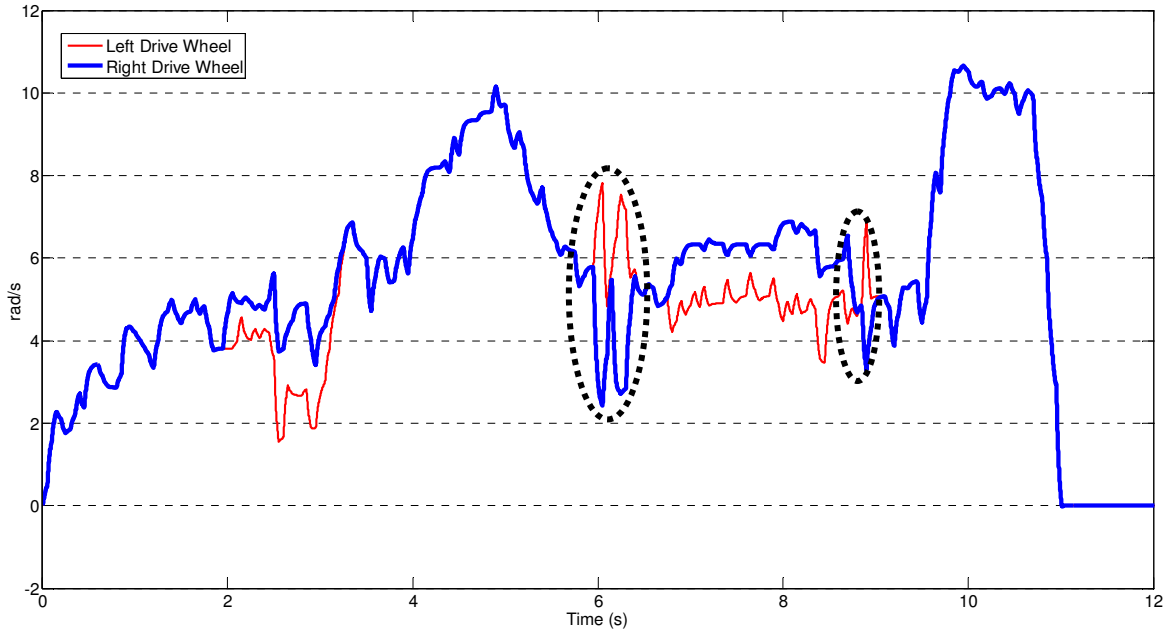

FIGURE 6.37: KINEMATIC MODEL DRIVE WHEEL ROTATIONAL VELOCITIES FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL

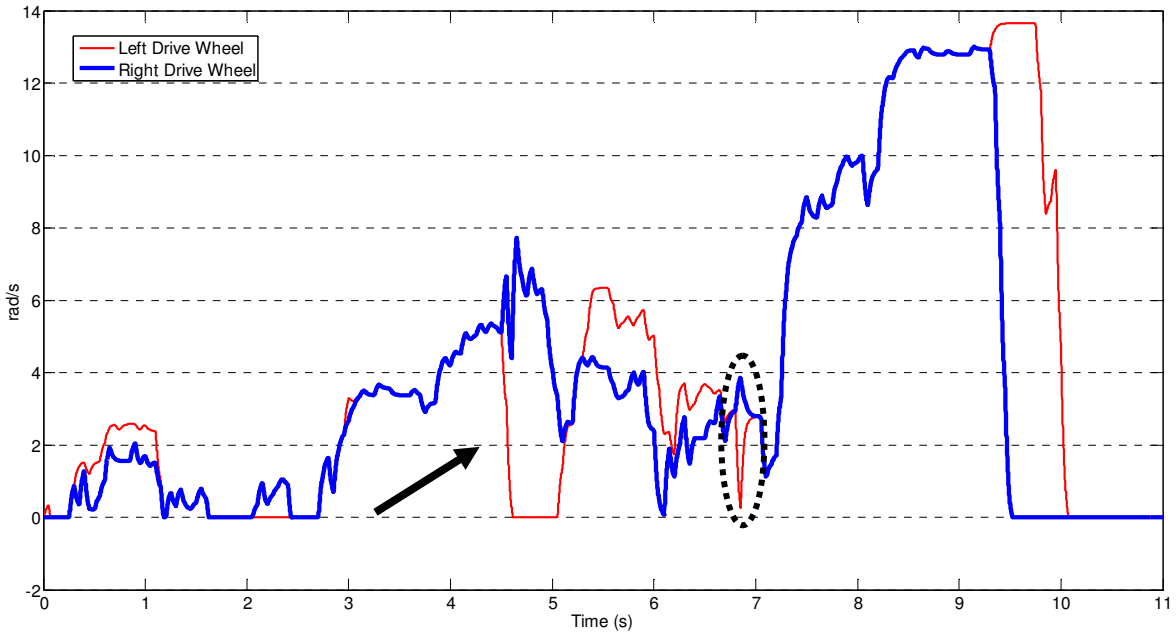

FIGURE 6.38: KINEMATIC MODEL DRIVE WHEEL ROTATIONAL VELOCITIES FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL

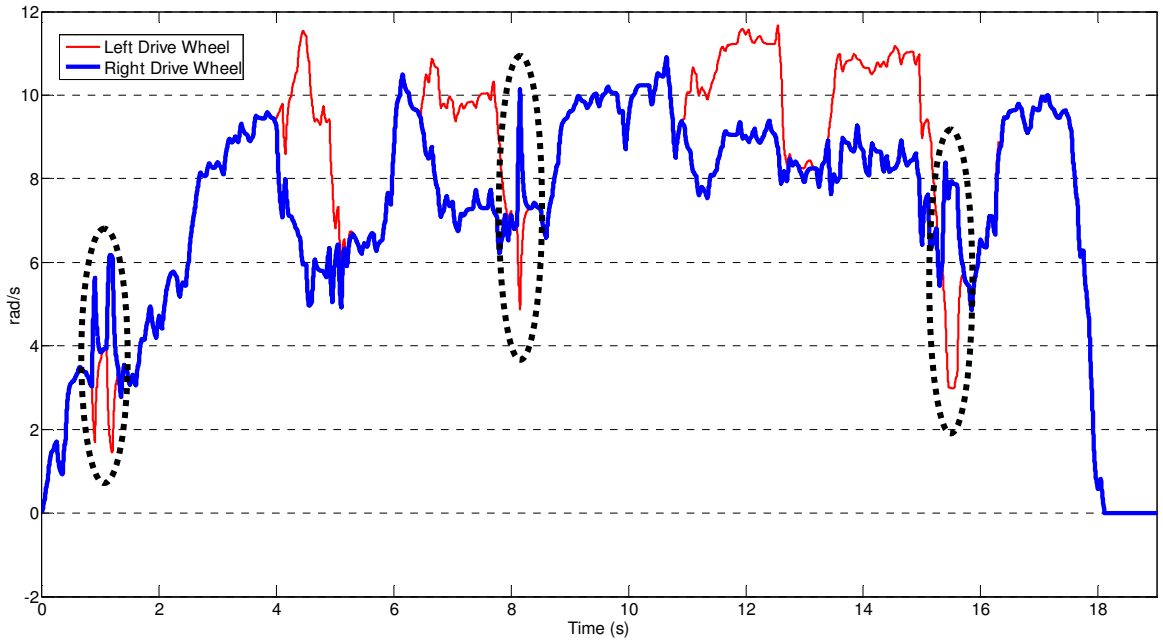

FIGURE 6.39: KINEMATIC MODEL DRIVE WHEEL ROTATIONAL VELOCITIES FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL

The circled areas are referred to as steps because at these points there is a difference between each drive wheels' input demands of 20% or greater. Furthermore, this large difference is immediately preceded and followed by a point where the wheel velocities are identical and therefore zero difference in input demand. Hence, the reference to a step input.

Tables 6.2 to 6.4 list the duty cycle demand inputs and their differences at each relevant time step in the model for each set of the defined data group. It should be noted that in Table 6.2 that in 2 of the steps, 5.950 to 6.000 and 6.150 to 6.250, the step rule has not been followed in that the following duty cycle inputs are not identical at the 6.000 and 6.200 second marks. However, for each time step following these sets, 6.050 and 6.300 respectively, the differences are zero and the final steps in each series are large enough to cause a yaw spike.

TABLE 6.2: KINEMATIC MODEL STEP INPUTS FOR DOGLEG RIGHT PROFILE WITHOUT LOSS OF CONTROL

Right Duty Cycle demand (%)	Left Duty Cycle demand (%)	Difference	Time Step (s)
25	53	28	5.950
25	56	31	6.000
25	49	24	6.150
27	55	28	6.200
28	51	23	6.250
28	53	25	8.850

TABLE 6.3: KINEMATIC MODEL STEP INPUTS FOR DOGLEG RIGHT PROFILE WITH LOSS OF CONTROL

Right Duty Cycle demand (%)	Left Duty Cycle demand (%)	Difference	Time Step (s)
34	11	34*	6.800

* NB: Recall that the Roboteq controller treats any demand less than 13% as zero

TABLE 6.4: KINEMATIC MODEL STEP INPUTS FOR OVAL RIGHT PROFILE WITHOUT LOSS OF CONTROL

Right Duty Cycle demand (%)	Left Duty Cycle demand (%)	Difference	Time Step (s)
46	20	26	0.850
48	20	28	1.100
70	35	35	8.100
60	36	24	15.350

Referring to Figure 6.38, the arrow points to an area where one would expect a correlated spike in the yaw rates to occur. However, the steps in the demand inputs for this region are constrained in the manner the several large steps from the 4.500 mark to the 4.900 mark are followed by a difference in the demand inputs of 19, less than the defining 20 point difference and not zero at the 4.950 mark. It should be also noted for the rotational wheel velocities, although the model runs at 1 kHz, changes are only registered at every 50 ms in the model results.

The implication of these step inputs is that the kinematic model does not factor in the mass moment of inertia of the wheelchair. In contrast, in the actual test-bed, such sudden short step inputs are muted by the chair's inertia. Hence, there is a need to be able to factor mass

moment of inertia into the kinematic model. This change would likely lead to a change in the OCR value as at present step inputs are under-damped and smaller inputs, the differences between the drive wheel duty cycle demands of less than 20, are over-damped. A means to institute the inertia effects directly into the Kinematic model would bring it into closer correlation than at present and provide greater accuracy and precision than currently available with the Dynamic model.

6.4 SUMMARY

An accurate model of a FWD wheelchair has been created. The PM DC drive motor model, which is a key component of the FWD wheelchair model system has also been fully validated. An initial dynamic model containing equations of motion with the correct variables and parameter values has proven to be difficult to fully validate. A simpler kinematic model has been validated, in particular, the velocity outputs match very well, but the model requires further improvement in yaw outputs. The testing process used to validate the models, the results of the testing, and the analysis that covers changes made to parameters, variables and models was described fully in this chapter.

Testing of the PM DC drive motor and entire FWD wheelchair system models were done separately. It was found that one of the coupled governing equations of the PM DC drive motor was missing a vital parameter not normally covered in the literature [28]. The calculation and value of the missing parameter, as well as how its inclusion allowed validation of the PM DC drive motor model, is also documented.

The initial dynamic model inherited from prior research contained an incorrect mass moment of inertia value that when corrected improved the profiles and values of the velocity outputs, but decreased the values of the yaw outputs. It was also discovered that a variable used to calculate the centripetal force experienced by a wheelchair in a turn was incorrect. Correct centripetal force calculation is vital to account for over-steer effects. However, an ideal equation to calculate the variable proved inaccurate and lead to the abandonment of the dynamics model.

An initial purely kinematic model was found to accommodate the effects of over-steer in a FWD wheelchair with a factor. Further improvement in the model is suggested by adding a mass moment of inertia factor. The addition of this new factor to the model would be to dampen control drive wheel velocity output spikes that are also described in this chapter.

7

CONCLUSIONS

The general need for stability control for all types of EPWs has been well documented, reviewed and summarised in this research. The need to extend stability control to electrically powered FWD wheelchairs has also been acknowledged by a well established health industry firm. The most effective means to provide this stability is through a feedback control system using one or more solid state sensors to provide the closed-loop control based on information about the real-time driving state of the wheelchair.

This research also established that feedback control can only be designed if the nature of the system dynamics to be controlled is understood. The most effective means to gain this understanding is to capture pertinent data to create and validate a dynamic system model. Modelling has also been shown to be an effective means to design feedback control systems in many engineering systems and a validated model will thus likely produce the most cost effective, safe, and efficient design process.

7.1 FWD WHEELCHAIR TEST-BED DEVELOPMENT

A fully functioning wirelessly controlled test-bed was successfully developed. The development of this new test-bed overcame the problems of restricted mobility and severe

sensor noise seen in the prior tethered prototypes by using wireless control and data transmission systems. Thus, prior unsuccessful attempts at wireless control were overcome providing a more robust and capable test bed system.

7.2 DATA COLLECTION

Data was successfully collated, saved and processed from the 8 sensors of the test-bed wheelchair. Three different types of sensors: solid state accelerometers and gyroscopes, and rotary encoders, supplied the required data. It was found that the rotary encoders and gyroscopes provided adequate information for analysis and validation of a FWD wheelchair system model. In particular, the drive wheel rotary encoders provided valuable velocity and distance information and a single gyroscope provided yaw rate data about the normal axis through the wheelchair's CoG. The limits of gyroscope data were established and, in particular, the direct rotational rate information was exceptional. However, positional data obtained through the mathematical integration of the yaw rate contained drift errors. Overall, the sensor system and data provided all the required information for system model development and validation, as well as for testing and evaluating novel control strategies.

7.3 DATA ANALYSIS

The data analysis revealed positive outcomes with regard to stability, such as the fact that the FWD wheelchair is more stable occupied than unoccupied. This specific result disproved a theory proposing that raising of the wheelchair CoG due to an occupant would increase over-steer. It was found that the increase in the wheelchair's mass moment of inertia due to an occupant had a greater effect that increased stability, although only

slightly. Finally, the main theory that over-steer is primarily caused by the CoG location behind the drive wheels with no counter-acting forces from the rear casters in a turn, was confirmed.

7.4 MODEL VALIDATION

Model validation relies on accurate governing equations, correct values for the parameters identified in the equations, and understanding of the processed validation data. A dynamic model that took into account mass moments of inertia and over-steer effects found in FWD wheelchairs was inherited from previous research and attempts were made to validate this model. Several modelling errors were identified in the governing equations and variables within those equations. However, even with corrections, reasonable results for this dynamic model were unobtainable.

A kinematic model that did not take in to account the effects of over-steer or mass moment of inertia was found to be far more accurate. A factor was added to this model that accounts for over-steer. The kinematic model has proven to be the simpler and more effective model, yet there is still room for improvement. The velocity output of the model has MSE values of 0.1018, 0.635, and 0.0306 when compared to the actual velocities of the defined data group of 2 right doglegs, the first without loss of control and the second with loss of control, and a right oval pattern used for model validation. The model's yaw outputs has MSE values of 0.6542, 4.5002, and 1.5249 when compared to the actual yaw rates in the same order as the velocity comparisons. It is believed that with the addition of mass moment of inertia effects that the MSE yaw values would improve dramatically.

The model of the PM DC motor and drive system that is critical to both the dynamic and kinematic models was fully validated. The model's output current and rotational velocities, for various different voltage inputs, MSE values stretched from a high of 0.131 to a low of 0.001 with the majority of errors near the lower end of the range when compared to the actual values.

In summary, this research has developed what appears to be a first attempt to create a valid model of a FWD wheelchair system to be used specifically for feedback control design. However, such a model can be extended to be used in Augmented Reality situations or help in mobile robot tracking problems or as part of a simulation training system. Furthermore, it appears that it would be a straightforward matter to extend the model for other electrical powered wheelchair types.

7.5 FEEDBACK CONTROL DEVELOPMENT

An un-validated dynamic model of the FWD wheelchair test-bed was successfully incorporated into an Emulator system that an engineer at Dynamic Controls Ltd (Christchurch, NZ) used to design a feedback control system utilising solid state gyroscopes. However, a validated model would have allowed a more efficient design process by fully testing the limits of control and allowing the tuning of the gains used in the designed feedback control system.

7.6 SUMMARY

Creation of a FWD wheelchair test-bed and collection of dynamic data aided in the analysis of two separate models, and in the understanding of the true nature of the over-steer phenomenon. During the FWD wheelchair model validation process, it was discovered that a dynamic system model could not be fully validated due to unmodelled non-linear effects. However, more positively, a simpler kinematic model was much more accurate and readily validated versus measured data. The dynamic FWD wheelchair model was also used in a commercial setting to design a feedback control system utilising a solid state gyroscope. However, a fully validated FWD wheelchair model would have improved this design process. Overall, a reasonable model of a FWD wheelchair has been created based on kinematic equations with a direct addition for over-steer effects found in the dynamics of FWD wheelchairs. This model would likely be a better foundation for further control design and analysis. Further improvement of this second model is desirable and should be readily attainable.

8

FUTURE WORK

8.1 FINAL MODEL DEVELOPMENT

8.1.1 Road Load

At present, counter torque due to road load has not been implemented in the kinematic model. Nor is the model adjusted to take into account driving on gradients. Therefore, adjusting Equation (4.13) by ignoring air resistance provides:

$$F_{RL} = \mu_r N + N \sin\theta$$

where θ is the angle of gradient. This change would most likely be integrated into the PM DC motor model through Equation (4.21) as follows:

$$k_t i = J\ddot{\theta} + B\dot{\theta} + F_{RL} RH(\ddot{\theta})$$

where $H(\ddot{\theta})$ is the Heaviside function. In a dynamic model, it is important to note that force only exists during accelerations. Thus the use of the Heaviside function in this equation.

8.1.2 Mass Moment of Inertia

At present, damping due to the mass moment of inertia has not been implemented in the kinematic model. This effect could most likely be applied to Equation (4.1) as follows:

$$I\ddot{\theta} + \dot{\theta} = \frac{r_{wheel}}{R_r}(\omega_L - \omega_R)$$

However, it is not clear if the moment of inertia, *I*, would be the actual wheelchair's value or the estimated value 5.5 kg•m^2. Further empirical experimentation would therefore need to be pursued to find the correct parameter value or method of implementation.

8.1.3 Over-steer Correction Factor (OCF)

As the mass moment of inertia is applied to kinematic model, the OCF will most likely need to be adjusted. However, it is also possible that another method of implementation would be needed. Again, empirical experimentation appears the way forward in this area to further improve the model and methods developed.

Appendix A

National Electronic Injury Surveillance System (NEISS)

Sample Case Detail

Glossary

PSU = Primary Sampling Unit (Hospital) Weight = Statistical Weight

Stratum = Size/type of hospital (S = Small, M = Medium, L = Large, V = Very Large, C = Children's Hospital)

Total Records: 2,943

CPSC Case #: 70347413 Treatment Date: 03/24/2007 PSU: 35 Weight: 15.1766 Stratum: V

Age: 80 Sex: 1 - MALE Race: 1 - WHITE Race Other:

Diagnosis: 58 - HEMATOMA Diag Other:

Body Part: 75 - HEAD

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 5 - OTHER PUBLIC PROPERTY

Products: 1707 - WHEELCHAIRS

Narrative: 80 YO MALE FELL FROM WHEELCHAIR AT NURSING HOME HEMATOMA SCALP

CPSC Case #: 70403770 Treatment Date: 03/24/2007 PSU: 73 Weight: 70.1094 Stratum: S

Age: 72 Sex: 2 - FEMALE Race: 1 - WHITE Race Other:

Diagnosis: 62 - INTER ORGAN INJURY Diag Other:

Body Part: 75 - HEAD

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 0 - UNKNOWN

Products: 1707 - WHEELCHAIRS

Narrative: PT FELL OUT OF A WHEELCHAIR STRIKING HEAD SUSTAINED A CLOSED HEAD INJURY WITH LACERATION TO SCALP

CPSC Case #: 70406140 Treatment Date: 03/24/2007 PSU: 24 Weight: 80.0746 Stratum:M

Age: 48 Sex: 1 - MALE Race: 1 - WHITE Race Other:

Diagnosis: 57 - FRACTURE Diag Other:

Body Part: 36 - LOWER LEG

Disposition: 4 - TREATED & ADMITTED FOR HOSPITALIZATION, HOSPITALIZED

Location: 0 - UNKNOWN

Products: 687 - BENCHES

1707 - WHEELCHAIRS

Narrative: RAN INTO BENCH IN ELECTRIC WHEELCHAIR FX TIBIA

CPSC Case #: 70407469 Treatment Date: 03/24/2007 PSU: 43 Weight: 70.1094 Stratum: S

Age: 28 Sex: 2 - FEMALE Race: 0 - N.S. Race Other:

Diagnosis: 57 - FRACTURE Diag Other:

Body Part: 92 - FINGER

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 1 - HOME

Products: 1707 - WHEELCHAIRS

Narrative: TINY AVULSION FX L 5TH FING/28YOF AT HOME FELL AND HURT FINGER ON WHEEL CHAIR.

CPSC Case #: 70438561 Treatment Date: 04/14/2007 PSU: 58 Weight: 15.1766 Stratum: V

Age: 85 Sex: 1 - MALE Race: 1 - WHITE Race Other:

Diagnosis: 57 - FRACTURE Diag Other:

Body Part: 79 - LOWER TRUNK

Disposition: 4 - TREATED & ADMITTED FOR HOSPITALIZATION, HOSPITALIZED

Location: 1 - HOME

Products: 1707 - WHEELCHAIRS

Narrative: FELL FROM WHEELCHAIR/HIP FX

CPSC Case #: 70438565 Treatment Date: 04/14/2007 PSU: 58 Weight: 15.1766 Stratum: V

Age: 51 Sex: 1 - MALE Race: 1 - WHITE Race Other:

Diagnosis: 52 - CONCUSSION Diag Other:

Body Part: 75 - HEAD

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 1 - HOME

Products: 1707 - WHEELCHAIRS

Narrative: FELL BACK IN WHEELCHAIR/CONCUSSION

CPSC Case #: 70455376 Treatment Date: 04/17/2007 PSU: 44 Weight: 69.0339 Stratum: L

Age: 77 Sex: 1 - MALE Race: 1 - WHITE Race Other:

Diagnosis: 62 - INTER ORGAN INJURY Diag Other:

Body Part: 75 - HEAD

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 1 - HOME

Products: 1707 - WHEELCHAIRS

Narrative: FELL OUT OF WHEELCHAIR ONTO SIDEWALK AT HOME. DX CHI

CPSC Case #: 70500837 Treatment Date: 04/17/2007 PSU: 76 Weight: 80.0746 Stratum:M

Age: 73 Sex: 2 - FEMALE Race: 2 - BLACK Race Other:

Diagnosis: 57 - FRACTURE Diag Other:

Body Part: 81 - UPPER LEG

Disposition: 4 - TREATED & ADMITTED FOR HOSPITALIZATION, HOSPITALIZED

Location: 1 - HOME

Products: 1807 - FLOORS OR FLOORING MATERIALS

1707 - WHEELCHAIRS

Narrative: LEFT DISTAL FEMUR FX-FOUND ON FLOOR AFTER FALLING OUT OF WHEELCHAIR. NO LOC OR HEAD INJURY.

CPSC Case #: 70514799 Treatment Date: 04/18/2007 PSU: 63 Weight: 80.0746 Stratum:M

Age: 87 Sex: 2 - FEMALE Race: 0 - N.S. Race Other:

Diagnosis: 53 - CONTUSION OR ABRASION Diag Other:

Body Part: 30 - SHOULDER

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 5 - OTHER PUBLIC PROPERTY

Products: 1807 - FLOORS OR FLOORING MATERIALS

Narrative: PT FELL OUT OF THE WHEELCHAIR TO THE FLOOR AT THE NURSING HOME LANDING ONTO LEFT SHOULDER CONTUSION

CPSC Case #: 70514851 Treatment Date: 04/20/2007 PSU: 63 Weight: 80.0746 Stratum:M

Age: 93 Sex: 2 - FEMALE Race: 0 - N.S. Race Other:

Diagnosis: 53 - CONTUSION OR ABRASION Diag Other:

Body Part: 82 - HAND

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 0 - UNKNOWN

Products: 1707 - WHEELCHAIRS

Narrative: PT FELL OUT OF A WHEELCHAIR SUSTAINED A CONTUSION TO RIGHT HAND

CPSC Case #: 70515046 Treatment Date: 04/27/2007 PSU: 53 Weight: 15.1766 Stratum: V

Age: 64 Sex: 2 - FEMALE Race: 1 - WHITE Race Other:

Diagnosis: 71 - OTHER OR NOT STATED Diag Other: NO INJURIES

Body Part: 87 - N.S./UNK

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 1 - HOME

Products: 1707 - WHEELCHAIRS

Narrative: NO INJURIES-FELL BACKWARDS FROM A WHEELCHAIR-@ HOME

CPSC Case #: 70515069 Treatment Date: 04/27/2007 PSU: 53 Weight: 15.1766 Stratum: V

Age: 49 Sex: 2 - FEMALE Race: 1 - WHITE Race Other:

Diagnosis: 53 - CONTUSION OR ABRASION Diag Other:

Body Part: 79 - LOWER TRUNK

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 1 - HOME

Products: 1707 - WHEELCHAIRS

Narrative: HIP CT-APPARENTLY FELL FROM WHEELCHAIR-@ HOME

CPSC Case #: 70525397 Treatment Date: 04/30/2007 PSU: 43 Weight: 70.1094 Stratum: S

Age: 69 Sex: 1 - MALE Race: 0 - N.S. Race Other:

Diagnosis: 53 - CONTUSION OR ABRASION Diag Other:

Body Part: 76 - FACE

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 1 - HOME

Products: 1707 - WHEELCHAIRS

Narrative: MULT FACIAL CONT/69YOM FELL OUT OF WHEELCHAIR AT HOME AT 3:30P. C/O NECK, FACE AND BACK PAIN.

CPSC Case #: 70533079 Treatment Date: 04/18/2007 PSU: 38 Weight: 15.1766 Stratum: V

Age: 84 Sex: 1 - MALE Race: 0 - N.S. Race Other:

Diagnosis: 62 - INTER ORGAN INJURY Diag Other:

Body Part: 75 - HEAD

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 5 - OTHER PUBLIC

Products: 1707 - WHEELCHAIRS

Narrative: FELL FORWARD OUT OF WHEELCHAIR AT NURSING HOME RESULTING IN CLOSED HEAD INJ

CPSC Case #: 70540625 Treatment Date: 04/30/2007 PSU: 48 Weight: 70.1094 Stratum: S

Age: 96 Sex: 2 - FEMALE Race: 0 - N.S. Race Other:

Diagnosis: 57 - FRACTURE Diag Other:

Body Part: 79 - LOWER TRUNK

Disposition: 4 - TREATED & ADMITTED FOR HOSPITALIZATION, HOSPITALIZED

Location: 5 - OTHER PUBLIC PROPERTY

Products: 1707 - WHEELCHAIRS

Narrative: PT FELL OUT OF WHEELCHAIR AT NURSING HOME-FX LEFT HIP

CPSC Case #: 70548634 Treatment Date: 05/12/2007 PSU: 38 Weight: 15.1766 Stratum: V

Age: 82 Sex: 1 - MALE Race: 0 - N.S. Race Other:

Diagnosis: 53 - CONTUSION OR ABRASION Diag Other:

Body Part: 82 - HAND

Disposition: 1 - - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 5 - OTHER PUBLIC PROPERTY

Products: 1707 - WHEELCHAIRS

Narrative: FELL OUT OF WHEELCHAIR AT NURSING HOME AND ABRADED HAND

CPSC Case #: 70550913 Treatment Date: 05/12/2007 PSU: 58 Weight: 15.1766 Stratum: V

Age: 89 Sex: 1 - MALE Race: 1 - WHITE Race Other:

Diagnosis: 53 - CONTUSION OR ABRASION Diag Other:

Body Part: 75 - HEAD

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 5 - OTHER PUBLIC PROPERTY

Products: 1707 - WHEELCHAIRS

Narrative: TIPPED OVER IN WHEELCHAIR AT NH/SCALP CT

CPSC Case #: 71039406 Treatment Date: 10/15/2007 PSU: 61 Weight: 15.8664 Stratum: V

Age: 78 Sex: 2 - FEMALE Race: 1 - WHITE Race Other:

Diagnosis: 57 - FRACTURE Diag Other:

Body Part: 37 - ANKLE

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 1 - HOME

Products: 1707 - WHEELCHAIRS

1884 - CEILINGS AND WALLS (INTERIOR PART OF COMPLETED STRUCTURE)

Narrative: PT WITH FRACTURED ANKLE, GOT CAUGHT BETWEEN WHEELCHAIR AND WALL

CPSC Case #: 71063143 Treatment Date: 10/19/2007 PSU: 26 Weight: 15.8664 Stratum: V

Age: 81 Sex: 2 - FEMALE Race: 1 - WHITE Race Other:

Diagnosis: 53 - CONTUSION OR ABRASION Diag Other:

Body Part: 36 - LOWER LEG

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 5 - OTHER PUBLIC PROPERTY

Products: 1707 - WHEELCHAIRS

Narrative: CONTUSION TO LOWER LEG WHEN STRUCK BY A LADY IN A WHEEL CHAIR

CPSC Case #: 71063147 Treatment Date: 10/19/2007 PSU: 26 Weight: 15.8664 Stratum: V

Age: 79 Sex: 2 - FEMALE Race: 1 - WHITE Race Other:

Diagnosis: 59 - LACERATION Diag Other:

Body Part: 75 - HEAD

Disposition: 1 - TREATED & RELEASED, OR EXAMINED & RELEASED WITHOUT TRTMNT

Location: 1 - HOME

Products: 1707 - WHEELCHAIRS

Narrative: LAC TO HEAD WHEN FELL FROM A WHEEL CHAIR AT HOME

Appendix B

The Figure B.1 shows how the FWD wheelchair system model was incorporated into an emulator that is run on a PC and used for feedback control design at Dynamic Controls.

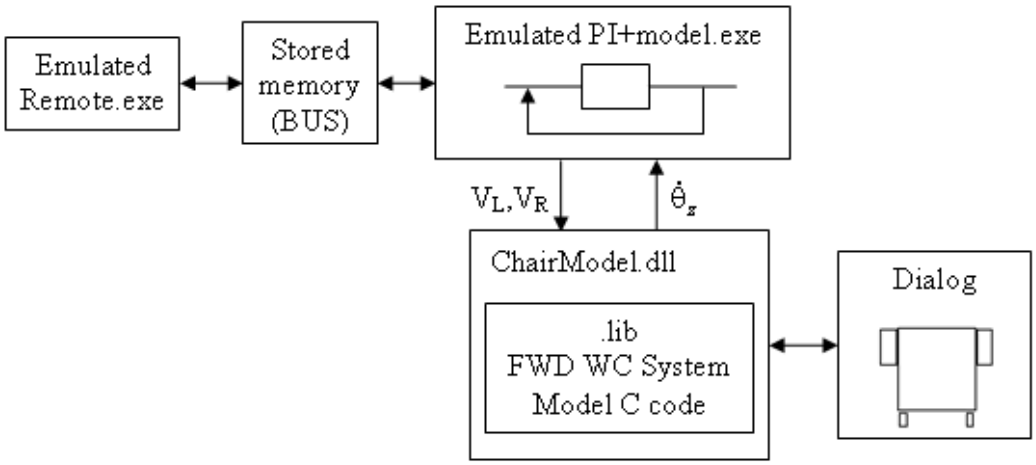

FIGURE B.1: INCORPORATION OF FWD WHEELCHAIR SYSTEM MODEL INTO EMULATOR FOR FEEDBACK CONTROL DESIGN

The Matlab Simulink Model of the FWD wheelchair was compiled into C language code and saved as a software library, .lib, within the ChairModel.dll or dynamic linked library. The dynamic linked library contained all functions necessary to communicate with the FWD wheelchair model in the library. The .dll was passed voltage demands for each drive motor from the Emulated Remote.exe via the Emulated PI+Model.exe. The rotational velocity around the wheelchair's vertical axis or yaw, was supplied to the Emulated PI+Model.exe. The .dll also fed the results of the drive forces due to the voltage demands produced by the drive motors, found in .lib, to a GUI. The GUI or in this case the Dialog, contained the plan view of a FWD wheelchair whose motion, due to the input drive forces from the .dll, could be visually observed.

The yaw information was also derived from the FWD wheelchair model in the library and this was passed via the .dll to the Emulated PI+Model.exe file or executable file. This executable file operated the designed feedback control system that adjusted the voltage demands accordingly. The Emulated Remote.exe ran the combined functions of a control joystick and motor controller. Inputs by the joystick were sent to and interpreted by the motor controller which applied the correct voltages, V_L and V_R, adjusted as necessary by the feedback control, to the drive motors in the lib.

Appendix C

The Figure C.1 shows the two axes of motion for a joystick, x and y. The x-axis represents turns and the y-axis represents forward and reverse motion. The 100 and –100 values at the end of each axes represent the limits of control demands.

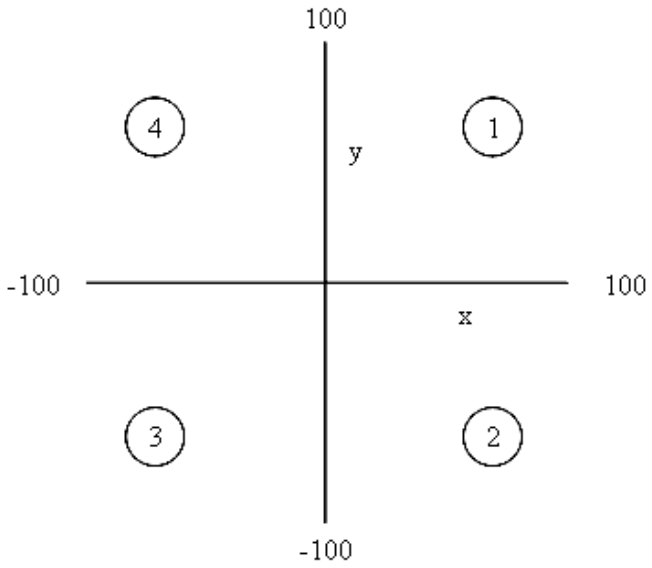

FIGURE C.1: RANGE OF MOTION OF JOYSTICK IN X AND Y AXIS

If the joystick was set as a remote to control the inputs to two drive motors the amount of control demand sent to the left and right drive motors is dictated by the following formulae:

$$\text{Left motor demand} = x + y \tag{C.1}$$

$$\text{Right motor demand} = y - x \tag{C.2}$$

Equations (C.1) and (C.2) would send the proper demands to each motor for joystick positions on either axes or in any of the four quadrants. For example, if the joystick was positioned for a sharp right turn, the 100 command input on the extreme right of the x-axis,

the Left motor would receive a demand of 100 + 0 or 100% full forward and the Right motor would receive a demand of 0 – 100 or –100% full reverse. Hence, in a differential drive set-up, the above input example would most likely cause the vehicle to spin on the spot to the right.

Appendix D

The programming of the PSoC comprised both assembly and C language code. Immediately below to page 161 is the assembly code. Starting towards the bottom of page 161 and going onto the final page in Appendix D is the C code.

```
;-------------------------------------------------------------------------

; Written:        Patrick Wolm
; Date:           13/07/2007
;
; This code is set up to take serial RS232 data in the form of unsigned 8
; bit integer packages, that have been converted to CMOS levels, through
; a UART module on the PSoC.  The input value represents a percentage
; duty cycle.
;
; The duty cycle value is then modified to adjust the pulse width of 2
; separate PWM modules in the PSoC device in order to send proper control
; values to a Roboteq controller in R/C Mode.  The limiting pulse widths
; are as follows:
;                 1 m/s       =     full reverse
;                 1.5 m/s     =     stop
;                 2 m/s       =     full forward
;
; The assembly code calls a c function, calculation.c, to modify the
; percentage duty cycle to a value which falls within the above limits.
; The PWM signals are then sent out to control the two motors of a front
; wheel drive wheelchair.
;-------------------------------------------------------------------------

include "m8c.inc"       ; part specific constants and macros
include "memory.inc"    ; Constants & macros for SMM/LMM and Compiler
include "PSoCAPI.inc"   ; PSoC API definitions for all User Modules

export _main
export _LEFT
export _RIGHT

area  bss (RAM)

COUNT:      BLK   1

_LEFT::
LEFT::      BLK   1     ; input duty cycle for Left wheelchair motor
```

```
_RIGHT::
RIGHT::           BLK   1     ; input duty cycle for Right wheelchair
                                ; motor

_RESULT1::
RESULT1::         BLK   1     ; resultant PW for left wheelchair motor

_RESULT2::
RESULT2::         BLK   1     ; resultant PW for right wheelchair motor

_LeftAdjusted::
LeftAdjusted::    BLK   1

_RightAdjusted::
RightAdjusted::   BLK   1

_Temp1::
Temp1::           BLK   4

_Temp2::
Temp2::           BLK   4

area  text (ROM,REL)

_main:

;---------------------------------------------------------------------------
; The Counter8 module is used to generate a clock for the UART to set a
; baud rate of 57.6 Kbaud, with the following settings:
;           Clock             48 MHz (SysClk*2)
;           Enable            High
;           Outputs           None
;           Period            103
;           CompareValue      78
;           CompareType       Less Than Or Equal
;           InterruptType     Terminal Count
;
; The above settings are used to calculate the baud rate as follows:
;           baud rate = 48 MHz / ((Period + 1) * 8)
;---------------------------------------------------------------------------

.start_all:

      M8C_EnableGInt

      call  Counter8_1_Start
      mov   A, UART_PARITY_NONE     ; no parity
      call  UART_1_Start
      call  PWM16_1_Start
      call  PWM16_2_Start

.initiate:
      mov         [COUNT], $2

.wait_for_data:
      call  UART_1_bReadRxStatus  ; wait for data to be received via UART
      and   A, UART_RX_COMPLETE
```

```
      jz    .wait_for_data        ; if nothing received continue to wait
      cmp   [COUNT], $1
      jz    .get_data_right

.get_data_left:
      call  UART_1_bReadRxData      ; read data from UART Rx buffer
                                    ; automatically into A
      mov   [LEFT], A               ; save in RAM for left motor
      dec   [COUNT]
      jmp   .wait_for_data

.get_data_right:
      call  UART_1_bReadRxData      ; read data from UART Rx buffer
                                    ; automatically into A
      mov   [RIGHT], A              ; save in RAM for right motor


.set_pulse_width:
      call  _Calculate        ; calling C function to perform signed
                              ; floating point operations
      mov   X, $0             ; move MSB into index register for new PW
      mov   A, [RESULT1]      ; move LSB into accumulator register for
                              ; new PW
      call  PWM16_1_WritePulseWidth ; set new pulse width for left motor
      mov   X, $0                   ; as above for right motor
      mov   A, [RESULT2]
      call  PWM16_2_WritePulseWidth
      jmp   .initiate         ; start again and wait for more inputs

.terminate:
    jmp .terminate


;-----------------------------------------------------------------------
```

```
#include "calculation.h"
//We need to tell the compiler that we are going to use some variables outside this file. We
// use the 'extern' keyword. The format for using extern is:  extern <type> RamLocation
//        extern - Keyword to tell the compiler the location is outside this module
//       <type> - Type is the C declaration (char,int) of the ram location - default is int
//                      Ex.     Var blk  1 equates to char
//                              Var blk  2 equates to int
//                              Var blk  4 equates to long
// RamLocation - This is the name of the ram location with the "_" we declared in
//main.asm

extern char  LEFT;              // Notice we dont have the "_", we just need it on the export
extern char  RIGHT;
extern char  RESULT1;
extern char  RESULT2;
```

```
char  Adjust = 0X64;            // decimal 100
char  Base = 0x96;              // decimal 150 representing 1.5 m/s PW
char  Range = 0x32;             // decimal 50 representing the range to max or min PW
extern signed char  LeftAdjusted;
extern signed char  RightAdjusted;
extern float  Temp1;            // temporary floating point values
extern float  Temp2;


//Here we declare our C Function. We do not use the _ even though we call it using the "_"

void Calculate()
{
        // In this function LEFT and RIGHT have 100 subtracted from their value, after
        // which they are divided by 100 to form a percentage.  Next is multiplication by
        // the Range and addition to the Base.
        signed char LeftSign = 1;                       // negative check
        signed char RightSign = 1;

        LeftAdjusted = LEFT - Adjust;
        RightAdjusted = RIGHT - Adjust;

        if(LeftAdjusted < 0)
        {
                LeftSign = -1;
                LeftAdjusted = LeftAdjusted * LeftSign;
        }

        if(RightAdjusted < 0)
        {
                RightSign = -1;
                RightAdjusted = RightAdjusted * RightSign;
        }

        Temp1 = (float)LeftAdjusted/(float)Adjust;          //must cast chars to float values
        Temp2 = (float)RightAdjusted/(float)Adjust;

        LeftAdjusted = (float)Range * Temp1;
        RightAdjusted = (float)Range * Temp2;

        RESULT1 = Base + (LeftAdjusted * LeftSign);
        RESULT2 = Base + (RightAdjusted * RightSign);
}
```

Appendix E

The graphical code to run the CompactRIO operation comprised both a real time VI and an FPGA VI. The real time VI, seen in Figure E.1, collated and formatted all the data from the various sensors and handled external serial communications. The FPGA VI captured and processed all the various wheelchair sensor inputs. Data from the gyros was taken straight through without further processing. Data for the rotary encoders was processed to register current count values for each time step before being sent through to the real time VI. The accelerometer data was also pre-processed before being sent through to the real time VI. The accelerometers output a PWM signal and the FPGA code was used to find the leading and trailing edge of each pulse in order to calculate the period of the PWM, which equated to acceleration and is represented by a voltage value. Only the first frame of each Accelerometer block of a 4 step process is shown.

The FPGA VI, separated into 2 separate figures for clarity, can be seen in Figures E.2 and E.3. However, all sensor data capture and processing occurs within the same "Sampling Frame". Frames ensure that processes are carried out in set progressive steps. While loops ensure that the processes continue constantly until the system is manually stopped or power to the CompactRIO is terminated.

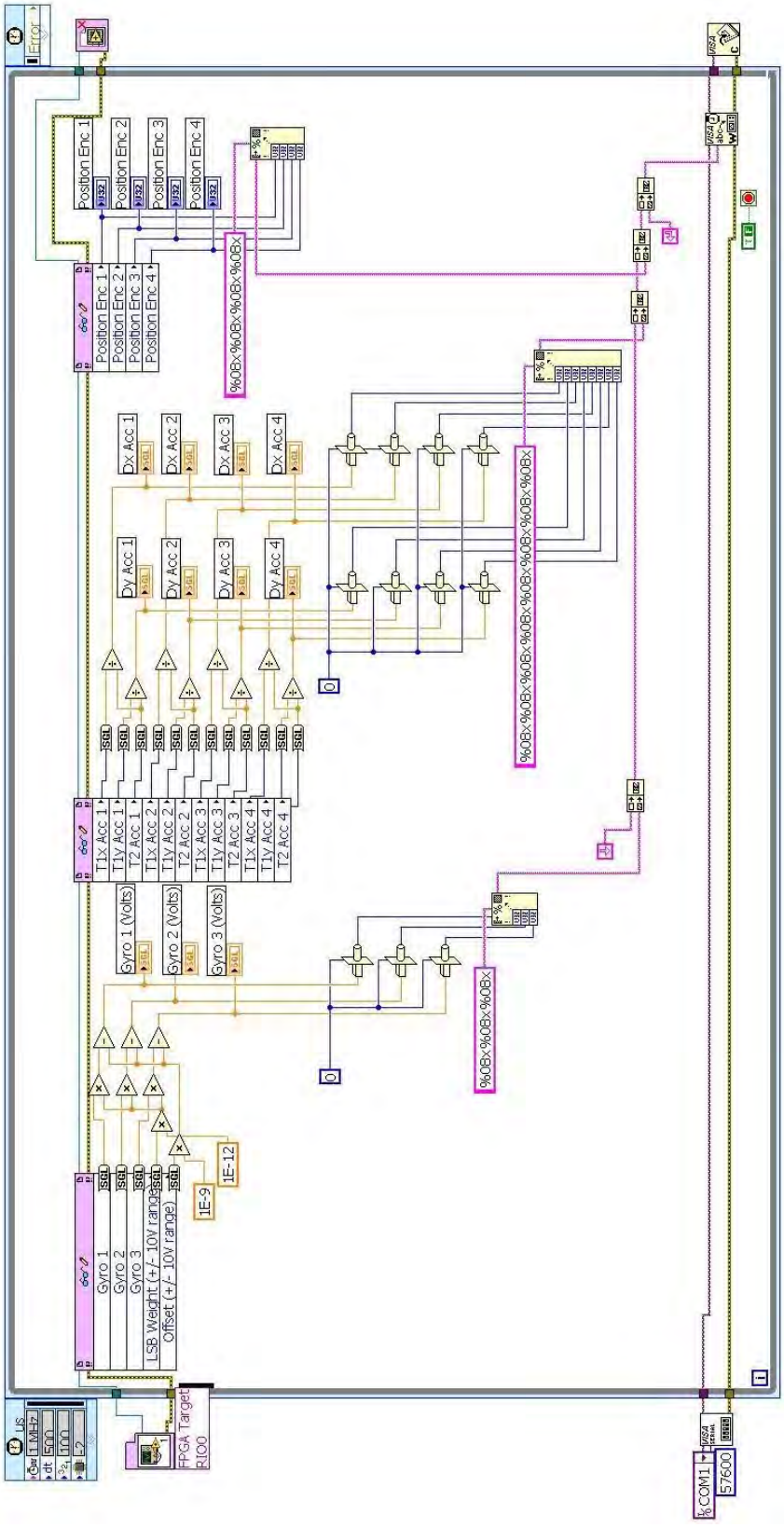

FIGURE E.1: REAL TIME VI THAT COLLATES SENSOR DATA FOR SERIAL TRANSMISSION FROM COMPACTRIO

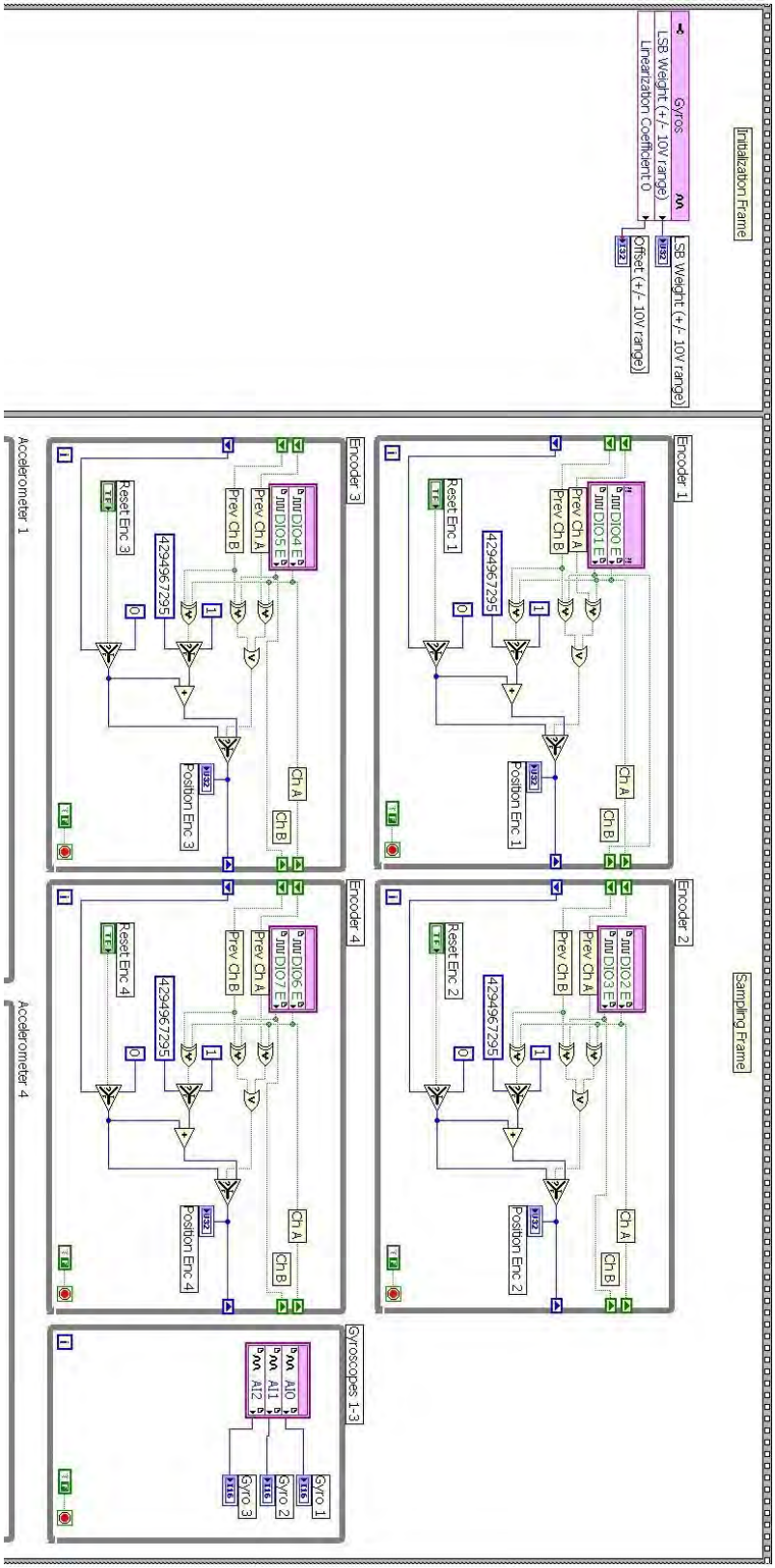

FIGURE E.2: FPGA VI SHOWING THE WHILE LOOPS THAT PROCESS THE ENCODER AND GYRO DATA BEFORE BEING PASSED TO THE REAL TIME VI

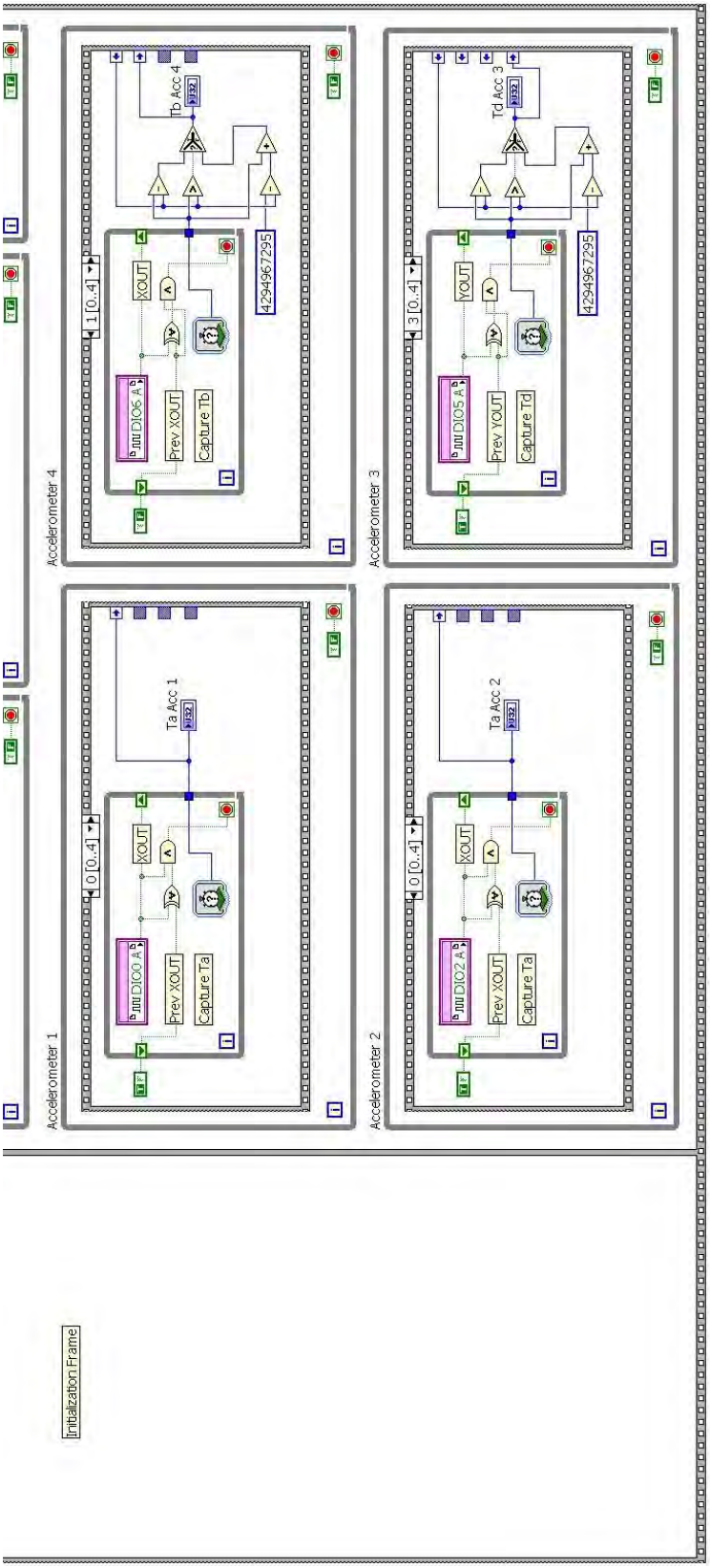

FIGURE E.3: FPGA VI SHOWING THE WHILE LOOPS THAT PROCESS THE ACCELEROMETER DATA BEFORE BEING PASSED TO THE REAL TIME VI

Appendix F

The m-files used to process and present the data can be separated into two parts. Part 1 of the data processing converts the raw data found in .csv files into basic voltages or count values that are the direct outputs of the solid state gyroscopes and accelerometers and encoders respectively. Then the data was changed into rotational velocity and acceleration values. Rotary encoder counts were further processed to provide basic rotational and linear distances. Part 2 of the data processing involved making plots of the rotational rate and acceleration values. Furthermore, the pre-processed encoder data was changed into drive wheel rotational and linear velocities. The linear wheel velocities were translated into overall wheelchair velocity to be plotted as necessary.

Part 1 M-files:

MumProgram3.m: written by John Oldridge, William Hanbury-Webber, Patrick Wolm

Main function that begins the raw data processing as follows:

MumProgram3(**'filename.csv'**)

The following external sub-functions are called from MumProgram3:

Extract_RIO4.m

ASCIIhex2dec.m

dechex2bits.m

hex2bits.m

bin2fpn.m

DecodedData4.m

Trajectory.m

MotorInputs.m

Finishes by saving fully processed data in a single file, **final_data.mat**.

Extract_RIO4.m: written by Patrick Wolm and Brendon Hale

Removes extraneous lines in data created by the DSpace system. Also reformats the data into its proper order that was re-arranged by the DSpace system. Separates out the sensor data recorded during the calibration period before each live test run with no motor inputs and saves calibration data into separate data file, **Zeroes.mat** that is opened in MumProgram3.m for further processing. Removes the recorded duty cycle control demands input during actual test-runs and saves into a separate data file, **Timestamp_DCs.mat** that is also opened in MumProgram3.m for further processing. Returns reformatted and extracted raw sensor data to MumProgram3.m directly.

ASCIIhex2dec.m: written by Patrick Wolm

Changes sensor data values to actual integer equivalents and returns processed data to MumProgram3.m. Data from the test-bed is transmitted in ASCII form that represents hexadecimal values. Therefore, the ASCII values are transformed into integers that represent hexadecimal values.

dechex2bits.m: written by Patrick Wolm

Function called from the **convert** function that is itself imbedded in MumProgram3.m. Changes individual sensor data, that is in integer format, into actual hexadecimal values. The hexadecimal values are added together and results in an unsigned integer. If the sensor data is from the encoders, the integer value is returned to MumProgram3.m and is the final real value of encoder counts. If the sensor data is from the gyroscopes the integer value is changed into a 32 bit binary string using the standard Matlab function, **dec2bin**, before being returned to the convert function in MumProgram3.m.

hex2bits.m: written by Patrick Wolm

Function called from the **convert** function that is itself imbedded in MumProgram3.m. Takes integer values for the accelerometers received via Extract_RIO4.m and changes each integer byte in an 8 byte 'word' into a 4 bit binary nibble using dec2bin. Then all 8 of the binary nibbles are concatenated together to create a 32 bit binary string before being returned to the convert function in MumProgram3.m.

bin2fpn.m: written by John Oldridge and William Hanbury-Webber

Function called from the **convert** function that is itself imbedded in MumProgram3.m. Takes all binary strings and changes them to floating point values that represent a voltage sensor reading from the accelerometers or gyroscopes. The m-file function is written using the IEEE Standard for Binary Floating-Point Arithmetic 754 protocol.

DecodedData4.m: written by John Oldridge, William Hanbury-Webber, Patrick Wolm

Takes the sensor data that has been changed in voltage values or integer counts via the convert function and transforms the data into actual rotational rates from the gyroscopes and accelerations from the accelerometers or count and distance changes from the rotary encoders. Also opens a data file, **Cal_Values.mat**, calibration values from the original Zeroes.mat file that had been processed through the convert function as well. The values from Cal_Values.mat are used in the calculations of the accelerations and rotational rates by providing a baseline and eliminating any noise in the results. Returns the decoded data to MumProgram3.m and saves further decoded encoder data in the form of linear and radial distance changes in data files **wheel_position.mat** and **radians.mat**.

Trajactory.m: written by John Oldridge and William Hanbury-Webber

Plots the path of the drive wheels and CoG of the wheelchair test-bed from an actual test-run in X and Y co-ordinates in the plane view. Decoded encoder data in the form of count changes is used to plot position and angle of orientation.

MotorInputs.m: written by Patrick Wolm

Takes the recorded duty cycle control demands saved in the Timestamp_DCs.mat file opened in MumProgram3.m and transforms the data that is in column format to row format. The transformed data is saved into a new file, **DCinputs.mat**, in its required format for use in the Simulink FWD wheelchair system model.

Part 2 M-files:

Forces5.m: written by Patrick Wolm

Takes the various .mat files created during initial processing by MumProgram3.m and plots results as needed. However, Forces5.m also calls several sub-functions, **velocities.m**, **Rescale2.m**, and **Velocities5.m** as necessary. Rescale2.m and Velocities5.m sought to smooth the various data by removing repeated values that covered multiple time steps. Repeated values occurred because the data was read faster than sensor updates transpired. The individual wheel rotational velocities data returned from Velocities5.m were then changed into overall wheelchair velocity. Forces5.m also saved separate .mat files when needed and served as the platform for plotting the FWD wheelchair Simulink model outputs.

velocities.m: written by Patrick Wolm

Takes the values in wheel_position.mat and radians.mat and converts them to linear and rotational velocities of the wheels for every 1 ms time step of the recorded data.

Velocities5.m: written by Patrick Wolm

Takes the values in wheel_position.mat and radians.mat and converts them to linear and rotational velocities of the wheels for every step change of the recorded data. The rotary encoder data tended to change in varying steps from 20 ms to over a second. When the velocity was calculated for each step change instead of every 1 ms, the velocity data was smoothed and gave a more accurate plot.

Rescale2.m: written by Patrick Wolm

Takes the values for all the gryo and accelerometer data found in final_data.mat and rescales the original 1 ms steps to actual step sizes between changing values in the data. This operation smoothes the gyro and accelerometer data to give a more accurate plot.

References

[1] Alva P, Hobson DA (1996), *Computer simulation of powered wheelchair electro-mechanical systems*, RESNA Proceedings, 215–216.

[2] Brown KE, Inigo RM, Johnson BW (1990), *Design, Implementation, and Testing of an Adaptable Optimal Controller for an Electric Wheelchair*, IEEE Transaction on Industry Applications, 26(6): 1144–1157.

[3] Collins TJ, Kauzlarich JJ (1988), *Directional instability of rear caster wheelchairs*, Journal of Rehabilitation Research and Development, 25(3): 1–18.

[4] Cooper RA (1993), *Stability of a Wheelchair Controlled by a Human Pilot*, IEE Transactions on Rehabilitation Engineering, 1(4): 193–206.

[5] Corfman TA, Cooper RA, Fitzgerald SG, Cooper R (2003), *Tips and Falls During Electirc-Powered Wheelchair Driving: Effects of Seatbelt Use, Legrests, and Driving Speed*, Archives of Physical Medicine & Rehabilitation, 84(12): 1797–1802.

[6] Ding D, Cooper RA, Guo S, Corfman TA (2004), *Analysis of Driving Backward in an Electric-Powered Wheelchair*, IEE Transactions on Control Systems Technology, 12(6): 934–942.

[7] Ding D, Cooper RA (2005), *Electric powered wheelchairs*, IEE Control Systems Magazine, 25(2): 22-34.

[8] Fattouh A, Dadam Y, Pham DT (2008), *MATLAB-Based 3D Dynnamic Model of a Powered Wheelchair*, [http://conference.iproms.org/matlab_based_3d_dynamic_model_of_a_powered_wheelchair], IPROMS Conference, unpublished.

[9] Gaal RP, Rebholtz N, Hotchkiss RD, Pfaelzer PF (1997), *Wheelchair rider injuries: causes and consequences for wheelchair design and selection*, Journal of Rehabilitation Research and Development, 34(1): 58–71.

[10] Han S, Choi B, Lee J (2008), *A Precise Curved Motion Planning for a Differential Driving Mobile Robot*, Mechatronics, 19(9): 486–494.

[11] Invacare International, SARL (2008), *Motorized Wheelchair*. EP1972486 (A1).

[12] Invacare Corporation, *G-Trac™ Technology* [online] http://www.invacare.com/cgi-bin/imhqprd/genericPage.jsp?s=0&contentPage=GTRAC (22 December 2008).

[13] Kirby RL, Atkinson SM, MacKay EA (1989), *Static and Dynamic Forward Stability of Occupied Wheelchairs: Influence of elevated footrests and forward stabilizers*, Archives of Physical Medicine & Rehabilitation, 70(9): 681–686.

[14] Kirby RL, McLean AD, Eastwood BJ (1992), *Influence of caster diameter on the static and dynamic forward stability of occupied wheelchairs*, Archives of Physical Medicine & Rehabilitation, 73(1): 73–77.

[15] Kirby RL, Thoren FAV, Ashton BD, Ackroyd-Stolarz SA (1994), *Wheelchair stability and manoeuvrability: Effect of varying the horizontal and vertical position of a rear-antitip device*, Archives of Physical Medicine & Rehabilitation, 75(5): 525–534.

[16] Kirby RL, Davidson CA, MacLeod DA (1998), *Mathematical model of lateral wheelchair stability*, Saudi Journal of Disability Rehabilitation, 4(3): 173–75.

[17] Leslie A, Clayton K, Grayling D, Joyce J (2005), *Stability Control of Wheelchairs using Solid State Accelerometers and Gyros*, ENME 438 Project Team Report, University of Canterbury, Christchurch New Zealand, unpublished.

[18] Michigan State University, *Advanced Biomechanics of Physical Activity (KIN 831), Anthropometry* [online] https://www.msu.edu/course/kin/831/Lecture%20%20Anthropometry.ppt (14 October 2008).

[19] National Center for Health Statistics (1995), *National Health Interview Survey on Disability, Phase I file (1994-1995).*

[20] Pinkney SJ (2001), *The Stability and Manoeuvrability of Powered Wheelchairs*. Ottawa: National Library of Canada = Bibliothèque nationale du Canada.

[21] Rentschler AJ, Cooper RA, Fitzgerald SG, Boninger ML, Guo S, Ammer WA, Vitek M, Algood D (2004), *Evaluation of selected electric-powered wheelchairs using the ANSI/RESNA standards*, Archives of Physical Medicine & Rehabilitation, 85(4): 611–619.

[22] Rizzoni G (2004), *Principles and Applications of Electrical Engineering*, McGraw-Hill, New York.

[23] Siegwart R, Nourbakhsh I, (2004), *Introduction to Autonomous Mobile Robots*, MIT Press, Cambridge, Massachusetts.

[24] Steidel RF, Jr (1989), *An Introduction to Mechanical Vibrations*, 3rd Edition, Wiley, New York.

[25] Sunrise Medical HHG, Inc (2006), *Active stability wheel chair based on positive angle sensors*. EP1686942 (A2).

[26] United States Census Bureau (2005), *Americans with Disabilities: Household Economic Studies*.

[27] Wang H, Grindle GG, Conner S, Cooper RA (2007), *An Experimental Method for Measuring the Moment of Inertia of an Electric Powered Wheelchair*, Proceedings of the 29th Annual International Conference of the IEEE EMPB, 22 – 27 August, 4798–4801.

[28] Wolm P, Anstis I, Oldridge J, Hanbury-Webber W (2006), *Stability Control of Wheelchairs using Solid State Accelerometers and Gyros*, ENME 438 Project Team Report, University of Canterbury, Christchurch New Zealand, unpublished.

[29] Wolm P, Chen XQ, Chase JG, Pettigrew W, Hann CE (2008), *Analysis of a PM DC Motor Model for Application in Feedback Design for Electric Powered Mobility Vehicles*, 15th International Conference on Mechatronics and Machine Vision in Practice, 2-4 Dec, Auckland, 640–645.